Real-time 3D-based
Virtual Eye Contact for Video Communication

DISSERTATION

zur Erlangung des akademischen Grades

Dr.-Ing.
im Fach Informatik

eingereicht an der
Mathematisch-Naturwissenschaftlichen Fakultät
der Humboldt-Universität zu Berlin

von
Dipl.-Inf. Wolfgang Waizenegger

Präsidentin der Humboldt-Universität zu Berlin:
Prof. Dr.-Ing. Dr. Sabine Kunst

Dekan der Mathematisch-Naturwissenschaftlichen Fakultät:
Prof. Dr. Elmar Kulke

Gutachter(innen):
1. Prof. Dr. Peter Eisert
2. Prof. Dr. Ralf Reulke
3. Prof. Dr. Thomas Sikora

eingereicht am: 27. Februar 2018
Tag der Verteidigung: 29. Mai 2019

Abstract

A major problem, that decreases the naturalness of conversations via video communication, is missing eye contact. While a person is looking on the display, she or he is recorded from cameras that are usually attached next to the display frame. During the last decades, many approaches have been proposed to solve this problem in various ways. These developments range from techniques that rely on pure image manipulation or basic view synthesis with stereo data to the tracking of 3D models, setups with specialized hardware or hybrid approaches that combine some of aforementioned procedures. With the advent of massively parallel computer hardware and in particular very powerful consumer graphics cards, it became possible to process many input views for real-time 3D reconstruction within a larger scale of communication setup where many cameras are involved. Here, a greater amount of input views mitigate occlusion problems and lead to a more complete set of 3D data that is available for view synthesis. In this thesis novel algorithms are proposed that enable for high quality real-time 3D reconstruction, the on-line alignment of photometric camera parameters and the automatic and user independent estimation of the eye contact cameras.

The real-time 3D analysis consist of two complementary approaches. On the one hand, a shape based algorithm and on the other hand, a patch based technique that evaluates 3D hypotheses via comparison of image textures. As texture data can be unreliable in case of homogeneous image regions and shape information only provides a coarse approximation to the real surface geometry, both approaches are combined in order compensate for these shortcomings. In addition, the algorithmic focus was on a massively parallel design that enables for efficient implementations on graphics hardware.

Once the 3D data is available it needs to be texture in order to render the eye contact view. Preparative to rendering, texture from multiple views needs to be aligned. Instead of post processing the recorded textures, a novel algorithm for photometric on-line adjustment of the camera parameters is proposed. The photometric adjustment is carried out iteratively in alternation with a 3D registration of the respective views. In this way the quality of photometric parameters is directly linked to the 3D analysis results and vice versa. Beside the preparation of a seamless multi-view rendering, the proposed adjustment of colorimetric properties also allows for a less expensive matching procedure for 3D analysis.

Based on the textured 3D data, the eye contact view is rendered. An important prerequisite for this task is the estimation of a suitable virtual eye contact camera. In contrast to many existing approaches that rely on a predefined fixed eye contact camera for all users, in this thesis a novel approach is formulated that enables for an automatic adaptation to arbitrary new users. Therefor, the eye contact camera is dynamically adapted to the current eye positions of the users. In this way, a virtual communication environment is created that allows for a more natural conversation. In particular it is possible to deliberately stop eye contact like in natural conversations and circumnavigate the remote conferee to the extend that is supported by the 3D data that underlies the rendering process.

Zusammenfassung

Ein Hauptproblem, das die Natürlichkeit von Unterhaltungen bei Videokonferenzen vermindert, ist fehlender Augenkontakt. Während eine Person auf den Bildschirm blickt, wird sie von Kameras aufgenommen, die sich normalerweise direkt daneben befinden. Im Laufe der letzten Jahrzehnte wurden viele Verfahren vorgeschlagen dieses Problem auf verschiedene Weisen zu lösen. Die Entwicklungen reichen von Techniken, die ausschließlich auf der Manipulation von Bildern beruhen oder einer elementaren Bildsynthese auf Basis von Stereo Schätzung, bis hin zur Verfolgung von 3D Modellen, Aufbauten mit Spezialhardware oder hybriden Ansätzen, die mehrere der genannten Prozeduren kombinieren. Mit dem Aufkommen von massiv paralleler Computer Hardware und ganz speziell den sehr leistungsstarken Spiele Grafikkarten ist es möglich geworden, viele Eingabeansichten für eine Echtzeit 3D Rekonstruktion in einem umfangreicheren Videokonferenzaufbau mit vielen Kameras zu verarbeiten. Eine größere Anzahl von Eingabeansichten mildert Verdeckungsprobleme ab und führt zu vollständigeren 3D Daten, die für eine Bildsynthese zur Verfügung stehen. In dieser Arbeit werden neue Algorithmen vorgeschlagen, welche eine hochqualitative Echtzeit 3D Rekonstruktion, die kontinuierliche Anpassung der photometrischen Kameraparameter und die benutzerunabhängige Schätzung der Augenkontaktkameras ermöglichen.

Die Echtzeit 3D Analyse besteht aus zwei komplementären Ansätzen. Auf der einen Seite gibt es einen Algorithmus, der auf der Verarbeitung geometrischer Formen basiert und auf der anderen Seite steht eine patchbasierte Technik, die 3D Hypothesen durch das Vergleichen von Bildtexturen evaluiert. In homogenen Bildbereichen können Texturen allerdings unzuverlässig sein und formbasierte Verfahren bieten lediglich eine grobe Annäherung an die wirkliche Oberflächengeometrie. Aus diesem Grund werden beide Ansätze kombiniert, um diese Nachteile wechselseitig zu kompensieren. Darüber hinaus lag der algorithmische Blickwinkel auf einem massiv parallelen Design, das eine effiziente Implementierung für Grafikkarten ermöglicht.

Nachdem die 3D Daten erzeugt wurden, ist es nötig sie zu texturieren, um eine Bildsynthese für die Augenkontaktansicht durchzuführen. Ein nötiger Schritt zur Vorbereitung der Synthese ist das Angleichen der Texturen von verschiedenen Ansichten. Anstatt eine Nachverarbeitung für die aufgenommen Texturen durchzuführen, wird die Anwendung eines neuen Algorithmus zur kontinuierlichen photometrischen Justierung der Kameraparameter vorgeschlagen. Die photometrische Anpassung wird iterativ, im Wechsel mit einer 3D Registrierung der entsprechenden Ansichten, ausgeführt. Auf diesem Weg ist die Qualität der photometrischen Parameter direkt mit jener der Ergebnisse der 3D Analyse verbunden und umgekehrt. Neben der Vorbereitung einer nahtlosen Bildsynthese aus vielen Ansichten erlaubt das vorgeschlagene Angleichen der photometrischen Kameraeigenschaften darüber hinaus, eine kostengünstigere Vergleichsprozedur für die 3D Analyse zu wählen.

Die Grundlage für eine Bildsynthese der Augenkontaktansicht sind texturierte 3D Daten. Eine weitere wichtige Voraussetzung dafür ist die Schätzung einer passenden virtuellen Augenkontaktkamera. Im Gegensatz zu vielen existierenden Ansätzen, die auf einer vordefinierten und festgelegten Augenkontaktkamera für alle Benutzer aufbauen, wird in dieser Arbeit ein neuer Ansatz formuliert, der die automatische Anpassung an beliebige neue Benutzer ermöglicht. Hierfür wird die Augenkontaktkamera kontinuierlich an die Augenposition der Benutzer angeglichen. Auf diesem Weg wird eine virtuelle Kommunikationsumgebung geschaffen, die eine natürlichere Kommunikation ermöglicht. Insbesondere ist es möglich, den Augenkontakt wie in einer natürlichen Konversation zu unterbrechen und die entfernte Person von verschiedenen Positionen zu betrachten, soweit die aufgenommen 3D Daten es unterstützen.

Danksagung

Zuerst möchte ich meinem Doktorvater Prof. Dr.-Ing. Peter Eisert für die wertvollen fachlichen Gespräche, seine Unterstützung und sein Feedback danken.
Ich danke auch meinen Vorgesetzten am Fraunhofer HHI Ingo Feldmann, Oliver Schreer und Peter Kauff für die Freiheit, meine Dissertation im Rahmen meiner dortigen Tätigkeiten verfassen zu können.

Außerdem danke ich meinen Kollegen Michael Reinhardt, Sascha Ebel und Markus Zepp für die fröhliche und produktive Arbeitsatmosphäre, die zahlreichen fruchtbaren Diskussionen und die großartige Zusammenarbeit - insbesondere möchte ich euch für die fortwährende Unterstützung bei den Laboraufbauten danken. Darüber hinaus möchte ich allen danken, die sich bereit erklärt haben Modell zu sitzen, um Testdatensätze zu generieren.

Meinen Dank möchte ich auch Jens Güther und Stefan Rauthenberg aussprechen für ihre umfangreiche und professionelle Unterstützung bei der Softwareentwicklung. Insbesondere Jens gilt hierbei zusätzlicher Dank für die vielen lustigen Momente während unserer Zusammenarbeit.

Für das aufmerksame und gründliche Korrekturlesen meiner Arbeit möchte ich vor allem meiner Freundin Diana und Wolf danken.

Diana gilt hier gleichzeitig mein allergrößter Dank für die immer währende Unterstützung und Ermutigung.

Contents

List of Figures

List of Tables

Nomenclature

Basic Notation

$\mathbb{R}$	Real numbers
$\mathbb{R}^+$	Positive real numbers
$\mathbb{R}^n$	n-dimensional vector space of real numbers
$\mathbb{P}^n$	n-dimensional projective space
$\mathbb{S}^n$	$n-$dimensional unit sphere
$\mathbb{S}^2_l$	$2-$dimensional lower hemisphere
Π	Patch prototype
Ω_I	Image plane of I that contains all valid image coordinates
$\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$	Canonical basis of $\mathbb{R}^3$
$\text{vec}(\cdot)$	The row wise vectorization operator
$\sim$	Equality up to a scalar factor λ, i.e. $\mathbf{a} \sim \mathbf{b} \Leftrightarrow \mathbf{a} = \lambda\mathbf{b}$
$\circ$	Hadamard product
$\times$	Cartesian product
$\text{arctan2}(\cdot\,,\cdot)$	Four-quadrant inverse tangent function
$\mathbf{0}_N$	$N \times 1$ vector with all elements 0
$\mathbf{x}$	Inhomogeneous vector, $\mathbf{x} = (x, y)^T \in \Omega$
$\mathbf{X}$	Inhomogeneous vector in 3D space $\mathbf{X} = (X, Y, Z)^T$
$\boldsymbol{\pi}$	Plane in 3D space $\boldsymbol{\pi} = (A, B, C, D) = (\mathbf{n}^T, D)$, with $\|\mathbf{n}\| = 1$
$I^t := I$	Texture image $I^t(\mathbf{x}) = (r, g, b)$
I^m	Mask image $I^m(\mathbf{x}) = \{fg, bg\}$, for foreground and background identification
I^d	Depth image $I^d(\mathbf{x}) = Z \in \mathbb{R}^+$
I^n	Normal image $I^n(\mathbf{x}) = (n_x, n_y, n_z)^T \in \mathbb{S}^2$
I^h	Hypotheses image $I^h(\mathbf{x}) = (p_0, \ldots, p_{M-1})^T \in \mathbb{R}^M$
$\mathcal{S}(\cdot, \cdot)$	Similarity measure $\mathcal{S} : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}$
$\mathcal{N}^L(\mathbf{x}, I^h, t)$	Hypotheses neighborhood as defined in section 3.2.3.3
$\mathcal{H}(\mathbf{x}, t)$	Hypothesis for coordinate value $\mathbf{x}$ at time t
$\langle \cdot, \cdot \rangle$	Scalar product

Camera Notation and Mappings

Internal camera parameters	$\mathbf{K} = \begin{pmatrix} f & s & u_x \\ 0 & \alpha f & u_y \\ 0 & 0 & 1 \end{pmatrix}$
3×3 rotation matrix	$\mathbf{R} = \begin{pmatrix} r_{00} & r_{01} & r_{02} \\ r_{10} & r_{11} & r_{12} \\ r_{20} & r_{21} & r_{22} \end{pmatrix}$
Camera center	$\mathbf{c} = (c_0, c_1, c_2)^T$
Projection matrix	$\mathbf{P} = \begin{pmatrix} p_{00} & p_{01} & p_{02} & p_{03} \\ p_{10} & p_{11} & p_{12} & p_{13} \\ p_{20} & p_{21} & p_{22} & p_{23} \end{pmatrix} \sim [\mathbf{KR} \,|{-}\mathbf{KRc}]$
Space point mapping onto the image plane	$\pi_{\mathbf{P}} : \mathbb{P}^3 \to \Omega, \mathbf{S} = (A, B, C, D)^T$ $\pi_{\mathbf{P}}(\mathbf{S}) := \begin{pmatrix} \frac{Ap_{00}+Bp_{01}+Cp_{02}+Dp_{03}}{Ap_{20}+Bp_{21}+Cp_{22}+Dp_{23}} \\ \frac{Ap_{10}+Bp_{11}+Cp_{12}+Dp_{13}}{Ap_{20}+Bp_{21}+Cp_{22}+Dp_{23}} \end{pmatrix}$
Homography mapping for inhomogeneous coordinates	$\mathcal{M}_{\mathbf{H}}(\mathbf{x}) := \begin{pmatrix} \frac{xh_{00}+yh_{01}+h_{02}}{xh_{20}+yh_{21}+h_{22}} \\ \frac{xh_{10}+yh_{11}+h_{12}}{xh_{20}+yh_{21}+h_{22}} \end{pmatrix}, \mathbf{H} = \begin{pmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{pmatrix}$
Orthogonal projection of point $\mathbf{X}$ onto the plane $\boldsymbol{\pi}$	$\text{proj}_{\boldsymbol{\pi}}(\mathbf{X}) = \mathbf{X} - \frac{\langle \mathbf{X}, \mathbf{n} \rangle + D}{\langle \mathbf{n}, \mathbf{n} \rangle} \mathbf{n}$

Acronyms

AAM	Active Appearance Model
FAP	Face animation parameter
MPEG	Moving Picture Experts Group
PU	Processing Unit
CPU	Central Processing Unit
GPU	Graphics Processing Unit
FPGA	Field Programmable Gate Array
fps	Frames per second
DSP	Digital Signal Processor
GPGPU	General-purpose computing on graphics processing units

FLOPS	Floating Point Operations Per Second
GFLOPS	GigaFLOPS
VH	Visual Hull
IBVH	Image Based Visual Hull
EPS	Exhaustive Patch-Sweep
IPS	Iterative Patch-Sweep
HRM	Hybrid Recursive Matching
L-HRM	Line-Wise Hybrid Recursive Matching
SGM	Semi-Global Matching
SSD	Sum of squared differences
SAD	Sum of absolute differences
NCC	Normalized cross correlation
PSNR	Peak-signal-to-noise ratio
MSE	Mean squared error
VGA	Video Graphics Array resolution (640×480)
QVGA	Quarter VGA (320×240)
UHD	Ultra-high-definition (3840×2160)
HD	High-definition (1920×1080)
QHD	Quarter High-definition (960×540)
QQHD	Quarter quarter High-definition (480×270)
RGB	Red, green and blue color space
HSI	Hue, saturation and intensity color space

1. Introduction

From the first public videophone service more than 70 years ago until today, the involved technology evolved rapidly. Video communication is publicly available in many places and the advent of mobile devices enables even for wireless video communication while being on the way. Consumer services like Microsoft Skype, Google Hangouts or Apple FaceTime are widely available at low prices or even free of cost. But also dedicated room-sized, high-end business solutions are available such as Cisco's TelePresence, Polycom's TPX and HP's HALO system. Depending on the communication quality and the overall feeling of *being there*, video communication can be considered as a substitute for face-to-face business meetings. In consequence, travel costs would be saved and emissions would be reduced. As there is a periodic increase regarding available bandwidth and computational power in the past years, one of the major improvements for both the consumer and the business video communication services was a continuous increase of video resolutions and display sizes. While a large or even life-size display with a crisp clear resolution provides more details of a remote conferee, the well-known *eye contact problem* becomes more prominent. During video communication, in general a person is looking onto the screen while being recorded from one or multiple cameras that are located beside the display. In consequence, the chat partners do not perceive eye contact and the larger the display, the more significant is the effect of the non eye contact perspective. An illustration of the eye contact distortion with a moderate sized $1045 \times 600\,\mathrm{mm}$ full HD display is provided in figure 1.1. It can be seen, that all views exhibit a considerable deviation from the eye contact view.

1.1 Solving the Eye Contact Problem

One of the most disturbing drawbacks that prevents natural conversation and therefore hinders the user acceptance is missing eye contact between the conferees [MBP95, Hoe12, BMB06, QM99]. While in point-to-point video communication persons perceive a discomfort of not being looked at, in multi-party setups, in addition, gestures might be misinterpreted and it might be difficult to assess who is talking to whom. Conceptually, three main approaches for solving the eye contact problem have been reported in literature. The first pursues an image-based algorithmic manipulation of the video stream for mitigating the effects of missing eye contact. The second relies on 3D data-based rendering of a virtual eye contact view and the third proposes the application of a teleprompter like hardware setup with a camera that is placed behind a half transparent screen. In this thesis, while focusing on point-to-point video communication, a solution to the eye contact problem is pursued

Figure 1.1: Eye contact distortion example for different camera mountings. The person was captured with eight cameras that were directly attached to the top, bottom, left and right display frame.

that can be conceptually assigned to the second, 3D data-based approach. In this context, the resulting objectives are the computation of real-time 3D data from multiple input views, the proper and automatic identification and update of the eye contact perspective and the constitutive rendering of a photo-realistic eye contact view. As video communication exhibits an application-inherent real-time constraint, one of the major challenges is a fast 3D analysis while maintaining a high quality of the resulting 3D data. These requirements are addressed with an efficient algorithmic design that allows for a parallel and scalable implementation in order to exploit the available processing power of modern graphics cards and many-core systems.

The presented algorithms have been part of various demonstrator setups for public exhibitions and lab presentations. For an overview and illustration, please refer to appendix A.3.

1.2 Contributions

In the following, the major contributions of this thesis are outlined briefly.

- A novel algorithmic approach for real-time multi-view 3D analysis is presented. It consists of a novel variant of an Image Based Visual Hull (IBVH) algorithm and a novel algorithmic framework for patch-based stereo estimation that is capable of including IBVH depth data as hypotheses seeds in order to improve the completeness of the results especially in case of unstructured homogeneous image regions. The focus of the IBVH enhancements was on a massively parallel algorithmic design that allows for the real-time computation of high resolution depth maps on parallel architectures like graphics hardware. The key components for the identification of potential line segment intersections and the required tests for 2D and 3D line intersections were replaced by efficient parallel counterparts. In addition, an implicit extension of the object is integrated for cases where the silhouette of the object is restricted by the im-

age border. The IBVH results are used individually or in combination with the novel framework for patch-based real-time stereo algorithm that is referred to as Iterative Patch-Sweep. It is based on the iterative evaluation and propagation of 3D hypotheses in terms of 3D patches. The parallel design of the algorithm allows for an efficient implementation on graphics hardware in order to maintain real-time constraints. As the hypotheses evaluation is based on a temporal interdependency with the previous frame, the algorithm exhibits a high rate of convergence for a consecutive input sequence. But even when starting from an uninitialized state, in general a converged state is reached after one or two iterations. In consequence, the algorithm enables for an efficient real-time computation of multiple stereo input streams. In particular, the hypotheses-based algorithmic structure allows the inclusion of shape-based depth information from IBVH or any other source of 3D data. The proposed combination of Patch-Sweep and IBVH mitigates the impact of mismatches for homogeneous image regions and constitutes a convenient alternative to the integration of depth sensors.

- A new algorithm for high accuracy depth driven photometric image registration is presented. The goal is to optimize photometric camera settings with respect to optimal depth estimation results. The algorithm is capable of a continuous fully automatic online adjustment of colorimetric camera settings and of the electronic off-line fine-tuning of photometric properties for recorded stereo sequences. The registration process is formulated in terms of an alternating energy minimization procedure, where the geometric and photometric registration energies are consistently incorporated into the same continuous energy functional. Since a depth-based geometric image registration and the photometric registration is pursued concurrently, in contrast to other techniques the quality of photometric registration is directly related to the performance of depth estimation and vice versa. In consequence, the proposed approach is perfectly suited to enhance the outcome of stereo and multi-view algorithms and it improves the visual experience of a constitutive view-synthesis. Regarding the implementation perspective, the presented registration method is designed with focus on parallelizability which allows for an efficient real-time implementation on graphics hardware.

- A novel algorithm for the fully automatic placement of the *eye contact camera* is introduced. As a prerequisite, the relative position of the cameras and the communication screens needs to be identified once during the setup of the communication hardware. Without any further prior user specific knowledge or manual interaction, the eye contact camera is computed based on 3D eye positions and line of sight information. In contrast to existing approaches, the proposed algorithm allows for a continuous update of the *eye contact camera* by evaluating a virtual current line of sight. The benefit is twofold. First, there is no single *sweet spot* where the user needs to remain in order to enable for an eye contact view. And second, there is no *Mona Lisa effect*, i.e. a conferee is able to circumnavigate his/her chat partner to some extend when moving.

1.3 Structure of the Thesis

The thesis is organized as follows.

Chapter 2 Background and Related Work Chapter 2 provides an overview on different state-of-the-art approaches for eye contact preserving video communication (section 2.1). In addition, the state-of-the-art that is related to the individual algorithms that are presented in this thesis is reviewed. In section 2.2, related work regarding shape-based and stereo estimation based 3D analysis is discussed. Section 2.3 reviews works on photometric image alignment and, finally, section 2.4 provides an overview on eye contact camera calibration.

Chapter 3 Real-time Multi-view 3D Analysis In chapter 3 the 3D processing is presented that underlies the generation of the virtual eye contact view. First, a novel Image Based Visual Hull variant that focuses on parallel processing is introduced in section 3.1. The experiments section 3.1.4 illustrates the results of the proposed approach and proves its algorithmic efficiency. And second, an iterative patch-based stereo estimation framework is proposed in section 3.2. The general algorithmic concept is explained in section 3.2.1. In order to provide a reference for quality and efficiency benchmarks, a non iterative version of the proposed stereo algorithm is introduced in section 3.2.2. The different building blocks of the iterative stereo estimation framework are presented in section 3.2.3. Finally, in section 3.2.4 an evaluation of various algorithmic variants, a comparison with state-of-the-art 3D estimation approaches and results on the combination with depth data from the Image Based Visual Hull are provided.

Chapter 4 Continuous Photometric Alignment Chapter 4 presents a depth-based algorithm for photometric image alignment. The proposed algorithmic formulation in terms of an energy minimization problem is discussed in section 4.1, while the optimization of the geometric registration is discussed in section 4.2 and the optimization for the photometric one in section 4.3. The combination of both into an alternating iteration scheme is discussed in section 4.4. Finally, in the experiments section 4.5, results of the achieved photometric registration are illustrated and the impact of colorimetrically aligned input streams on the 3D analysis that is presented in section 3.2 is evaluated.

Chapter 5 Eye Contact Provision Chapter 5 provides a detailed overview on the geometric constraints for eye contact provision and a fully automatic algorithm for the calibration of the eye contact camera. Section 5.1 covers geometric background for virtual eye contact provision, while the placement of the eye contact camera is discussed in section 5.2. Rendering results of various eye contact views are presented in section 5.3.

Chapter 6 Summary and Conclusion Finally, in chapter 6, the presented work is recapitulated and an outlook regarding potential extensions and future applications is provided.

1.4 Research Publications and Patents

In accordance with section 6, article 2b) of the doctorate regulations of the faculty of natural sciences at the Humboldt University of Berlin, parts of this thesis were presented at the following international conferences and workshops

- IEEE International Conference on Image Processing (ICIP) 2009 [Wai+09], 2011 [Wai+11], 2012 [Wai+12], 2013 [Wai+13] and 2016 [Wai+16]
- BBC 3D Processing Workshop 2008 [Wai09]
- International Symposium on Vision, Modeling and Visualization (VMV) 2011 [WFE11]
- SPIE Real-Time Image and Video Processing 2011 [WFS11]

or were patented [13b, 13a, 13c, 17]. These publications and patents are the foundation of this thesis, which incorporates them under the presented approaches for eye contact preserving video communication, together with algorithmic extensions, a more detailed algorithmic description, and updated results.

2. Background and Related Work

This chapter will give an overview on state-of-the-art approaches for eye contact preserving video communication and to the state-of-the-art that is related to the individual algorithmic developments of this thesis. In addition, the work of this dissertation is situated within the reviewed state-of-the-art.

2.1 Eye Contact Preserving Video Communication

In the following, integrated work on approaches for the solution of the eye contact problem is reviewed. This includes specialized solutions for the eye contact problem, but also free viewpoint telepresence approaches. The latter can be considered as generalized solution to create novel views including the eye contact perspective. From an algorithmic perspective, some of the reviewed works pursue very different approaches compared to this thesis, while others are conceptually similar. Basically, there are three major concepts that can be found in the literature. The first proposes the rendering of virtual eye contact views based on 3D data that is estimated from multi-view cameras, acquired from depth sensors or extracted from 3D models. The second pursues a subtle manipulation of camera images in order to improve the subjective feeling of being looked at. Here, the scene perspective is not corrected or only partially corrected. And the third relies on the application of a special hardware setup without any further image processing steps. In general, two options are reported. On the one hand, the direct recording of the eye contact view for a fixed spatial position is proposed. Here, teleprompter like half transparent screen techniques are used. On the other hand, the application of a *spatially faithful* camera and display configurations is pursued that mitigate the effect of missing eye contact without exactly solving it. Regarding these three major concepts for generalized spatial positions, only 3D data-based approaches facilitate the flexibility for a perspectively correct eye contact provision. In this context, approaches that are founded on one of the two other concepts have to be considered as partial solutions or approximations.

2.1.1 Virtual View Synthesis from 3D

Provided that sufficient 3D data is available, eye contact can be established by the synthesis of a novel view for a virtual eye contact camera. Here, the most challenging part is the generation of the required 3D data in real-time and with adequate quality. The reported work for this task can be categorized into the application of (multi-view) stereo or 3D from

shape techniques, 3D acquisition from depth sensors and hybrid approaches that combine both or include 3D models to aid view synthesis or image manipulation.

2.1.1.1 Multi-view

Stereo-based view synthesis has a long tradition in the field of eye contact corrected video communication. Seminal work on this approach was already published in the early 1990s [OLC93]. Here, a camera was attached at the top and at the bottom of the screen. The disparity maps for the interpolation of the centered eye contact view are computed with a dynamic programming stereo algorithm [Cox+92]. Subsequent work addressed the extension to a three camera configuration [LBW95]. One camera is placed at the top of the screen, one at the left side and one at the right side. For this purpose, a specialized trifocal stereo algorithm was developed with an additional focus on occlusion handling and smoothing of the temporal disparity field. While the concept of stereo-based eye contact was established, other authors worked towards improved real-time performance with dedicated hardware for disparity estimation [Ohm+98]. Here, a point-to-point eye video communication demonstrator for eye contact correction was set up with multiple FPGAs and DSPs as key hardware components. For each side a single wide baseline stereo configuration was selected as input for 3D processing. A similar concept for eye contact correction with dedicated hardware is introduced in [KS02] and a complementary technical description is provided in [LH02]. In contrast to [Ohm+98], a multiprocessor board that can be attached to standard PC hardware is used for disparity estimation. Additionally, the 3D processing of input from four cameras is supported in order to allow for a greater perspective change while focusing on a scalable multi-party scenario within a shared virtual table environment. As the computational capabilities of PC hardware increased, software-based solutions became feasible. Based on a novel dynamic programming approach for disparity estimation, a solution for the eye contact problem was proposed in [Cri+03]. Here, a stereo configuration with one camera at the left and one at the right of the display was used. Including horizontal view synthesis, the authors reported an overall performance of one frame every two seconds for QVGA input. During the same time, a demonstrator of a free viewpoint telepresence solution was set up [Gro+03]. Unlike the previous literature that has been reviewed in this section, the authors of this system do not use stereo processing, but rely on an Image Based Visual Hull (IBVH) algorithm [Mat+00] for 3D data generation. Depending on the adjusted level of detail, five to nine frames per second have been achieved. The hardware configuration for 3D processing consists of 16 cameras and 17 PCs. One PC was used for the silhouette extraction of each video stream, and one for the IBVH computation. A more complex stereo-based software solution was published in [Tsa+04]. The authors propose a computationally expensive disparity field initialization based on adaptive window template matching and a globally optimized post-processing. In order to allow for real-time processing, a fast algorithm for disparity field updates was developed. Shortly after, a more specialized telepresence system for the medical domain was developed [Wel+05]. It uses an early implementation of a graphics card-based Plane Sweep algorithm for depth estimation on multiple views together with a view-dependent pixel coloring for virtual view rendering

[Yan+04, Yan03]. The rendering results are presented on an autostereoscopic display. In the advent of GPGPU processing, specialized works on graphics card-based solutions for eye contact correction have been reported. In a series of publications [Dum+08, Dum+09, Dum+10] the authors describe the evolution of a system that uses multiple GPU-based stereo Plane Sweeps for depth estimation. During the same time, the authors of [CM09] targeted towards commercial set-top boxes for the mass market as an integrated solution for eye contact correction. Here, FPGAs are used to compute disparity fields from stereo input and perform post-processing and view rendering.

2.1.1.2 Depth Sensors

As early versions of time-of-flight (ToF) cameras became available, it was still difficult to integrate them into eye contact preserving video communication setups. While they have been designed for industrial application, the supported resolutions were rather low compared to color cameras, and the acquired depth information exhibit a significant noise level. However, after the first consumer products for depth sensing like the Microsoft Kinect came to the mass market, a very active research started to integrate these devices into existing multi-view approaches. A great advantage compared to industrial ToF cameras was the much higher resolution, the integrated color camera and the significantly lower price. Regarding telepresence, one of the first works on the integration of Kinect depth sensors was published in [MF11a]. In this work, the authors were focused on covering a whole room as a telepresence area. For that purpose, eleven Kinect devices were distributed and their output was post-processed and fused in order to provide a live reconstruction of the room for free viewpoint rendering. In a follow-up work [MF11b], the authors apply their technology in a *desktop scale* environment. Here, the acquired depth images from multiple Kinect cameras are used as input for a simple mesh generation algorithm. In order to provide consistent textures for rendering on a stereoscopic display, a multi-view color matching algorithm was proposed to adjust color settings from different cameras. Additionally, as overlapping fields of view cause interferences between multiple Kinects, different solutions to the interference problem were discussed. With further improvements of their multi-Kinect telepresence setup, the authors extend their system with eye tracking capabilities in order to adapt the rendered view of an autostereoscopic display with respect to the current head position [Mai+12], switch to a volumetric representation of 3D data for improved rendering quality [MF12] and integrate an optical see-through head-worn display [Mai+13]. While the works on depth sensors that have been covered so far are focused on telepresence, they do not directly address the eye contact problem. In this context, another room-sized multi-Kinect approach was set up in order to ameliorate the missing eye contact [Dou+12]. It uses a mixture of Kinect devices, panorama cameras and personal cameras together with a life size wall display. While Kinect is used for user segmentation, the panorama cameras provide a complete view of the room and the personal cameras provide an approximate eye contact view. But also small scale solutions have been reported. In [Kus+12a], a single Kinect together with one Point Grey Grasshopper camera is used for eye contact correction. Here, the additional color camera serves as a replacement for the

low quality Kinect color camera. The depth data from Kinect is processed according to [Kus+11] and registered with the texture in order to enable for eye contact view rendering. Another small scale approach with two Kinect cameras was proposed in [Kje+14]. Based on a voxel representation, both depth streams are combined similarly to the procedure that is presented in [New+11]. Afterwards, ray tracing is used to render the desired eye contact view.

2.1.2 Image Manipulation

The acquisition and the processing of 3D data is a computationally demanding task. In order to reduce the algorithmic complexity, pure monocular image-based approaches were proposed for approximate eye contact correction. In [JD02], a local image manipulation within the eye region was introduced. Based on eye tracking, the relevant image parts that need to be warped for eye contact correction are identified. A more extensive transformation of the whole image is proposed in [Yip05]. To author presents a two staged approach. First, an affine image transformation is used to adapt the perspective of the face. Second, based on an eye model, the eye region is changed by image warping. A similar procedure was presented in [SR11]. However, there is no separate correction of the eye region, but a homography-based warp of the whole image. Here, the homography is chosen in order to compensate for the displacement angle between the real camera and the eye contact camera.

2.1.3 3D Models and Hybrid Approaches

In this section, works are reviewed that combine different techniques for eye contact provision. Here, the building blocks are auxiliary 3D models and the already reviewed multi-view, depth sensor and image manipulation approaches. Early works on the combination of monocular image manipulation and 3D model aided perspective transformation of the head was reported in the context of Microsoft's GazeMaster project [GTZ99, Gem+00, GZ02]. Based on the tracking of interest points in the head region, a generic rigid 3D face model is positioned and used for perspective change with respect to the eye contact view. During the same time, a similar approach was published in [CKJ02]. However, instead of interest point tracking, a reference image is used as target for a registration of the current face image. Subsequently, a 3D face model together with the registration information is used for the synthesis of the eye contact view. A combination of personalized 3D face models, sparse stereo processing and head pose tracking is proposed in [YZ04, YZ02]. The 3D face models need to be precomputed. Together with the tracked head pose, the 3D model is used as initial result. In a next step, this result is refined with results from stereo and template matching. The resulting set of 3D points is triangulated in image space via Delaunay triangulation in order to receive a mesh representation of the conferee that can serve as input for eye contact view rendering. While also using a 3D model for geometry warping, instead of including depth information, the authors of [ER06] propose to estimate the face animation parameters (FAPs) that are defined in MPEG-4 in order to capture the facial dynamics. The required texture information for view synthesis is interpolated from an image cube that contains a limited number of prerecorded views. In a more recent work, the application

of Active Appearance Models (AAM) was proposed in order to enable for deformable 3D models with a compact representation [Bri+09]. Two goals were pursued. On the one hand, the authors target on increased naturalness of the rendered results and on the other hand, they want to enable their approach for low bandwidth connections. Based on the upcoming Kinect devices, the authors of [Fle+14] proposed the registration of different personalized 3D head models. Here, for each user various high resolution 3D head models with different facial expressions are captured with the KinectFusion algorithm [New+11]. According to the current facial expression during video communication, the adequate pre-captured 3D models are used for registration to the current depth data from the Kinect device. The eye contact view is subsequently rendered on basis of the registered high resolution 3D face models. In [ZYX11], a 3D sensor is used complementary to traditional stereo processing. In order to produce a more complete and robust result, the depth data from both sources are fused and used for eye contact view synthesis. An image manipulation approach based on Kinect depth data was reported in [Kus+12b]. Here, the perspective for the face area of the image is partially changed with respect to the eye contact view. For this purpose, the facial region is identified via a state-of-the-art 2D face tracker [SLC11]. Then, based on the Kinect depth data, the face region is warped and the occurring insertion seams are optimized for the lowest visual impact.

2.1.4 Specialized Communication Setups and Recording Hardware

The approaches for eye contact provision that have been discussed in the previous sections all rely on an algorithmic processing of the input views. In the following, work on the design and setup of specialized communication configurations and recording hardware is reviewed. In literature, two major approaches for non-algorithmic (approximate) eye contact solutions can be found: Teleprompter-like techniques with half transparent screens and *spatially faithful* camera and display configurations that mitigate the effect of missing eye contact without exactly solving it. A basic approach for improved gaze communication in multi-party configurations was introduced in [TR00]. In order to preserve gaze direction, each participant needs to have one separate device with integrated camera and display for each remote participant. Each of these devices shows one remote side and transmits its view to the respective remote user. In addition, the devices are arranged in order to simulate a real conversation situation with consistent spaces and to perceive approximate eye contact. In their work on Group Video Conferencing, Nguyen and Canny coined the definition of *spatial faithfulness* [NC05]. While direct eye contact is not demanded, the definition of *spatial faithfulness* includes gaze awareness for all participants. Similar to [TR00], the goal is to preserve spatial arrangement of the participants, but to use only one big screen for all of them. The authors propose a multi-party setup that supports three persons per remote side. Each system consists of three cameras, three projectors and a lenticular sheet. In this way, each participant can be recorded by an individual camera and perceive its individual view via one of the projectors. While all projectors target the same screen, the views are separated by the lenticular sheet. An approach for a one-to-many communication scenario with unidirectional eye contact is reported in [Jon+09]. Here, a structured light-based 3D

scanner is used to reconstruct a person in real-time. The resulting 3D data is transmitted to many remote sides and rendered on an autostereoscopic 3D display with respect to the correct eye contact perspective.

The works that were reviewed in this section so far aimed at spatial consistency and gaze awareness or provide unidirectional eye contact. In the following, work is discussed that focus on fully bidirectional eye contact based on half transparent screens. Here, the key concept is to place one or more cameras behind the viewing screen of the conferee. As the cameras are able to see through the half transparent screen, eye contact can be provided if the conferee is sitting at the resulting *sweet spot*. An early work on this approach was published in the context of the MAJIC system [OTM96, Oka+94]. The authors propose a configuration for a conference system with three remote sides and one person on each side. The eye contact view is recorded with one camera that is positioned behind a Contra Vision screen while the remote conferees are projected in front of the screen. A similar concept was proposed in [KK06]. However, instead of using a Contra Vision screen, a half transparent screen is used while the camera and the projector are configured for alternating operation. Once the camera records an image, a black frame is projected, and while the remote stream is projected, there is no camera capturing. A more advanced screen technology is used in [Reg+12]. While the camera is placed like in the MAJIC system, a Holographic Optical Element (HOE) based screen is applied in combination with back projection of the remote video stream. Here, two filters that are oriented orthogonally to each other are placed in front of the camera and the projector in order to prevent the camera to capture light that is emitted from the projector. A single camera requires the conferee to sit still at one fixed position in order to maintain eye contact. In order to relax the seating position of the conferee, the authors of [Ver+03, VWS02] increase the number of cameras behind the screen to three. Based on an eye tracker, the view among the three cameras that is closest to direct eye contact is selected for transmission to the remote side. An even more flexible concept regarding the seating position of the conferees was pursued within the European project 3DPresence [FP708]. The works of the projected were focused on a three party environment with up to two conferees per party. By exploiting the viewing cones of lenticular 3D screens, a separate eye contact view is provided for each of the two local conferees. Instead of directly recording the eye contact view, eye contact is established by view synthesis that is based on a multi-view camera setup. While there have been multiple scientific demonstrator setups in the last years, first versions of transparent screen based commercial solutions emerged on the market [DVE].

2.2 3D Analysis

In this thesis, 3D data is used for the rendering of virtual eye contact. The extraction of 3D information from multiple images or multiple video streams is a very active topic in the area of Computer Vision. This section concentrates on the work that is closely related to the algorithmic developments of this thesis. For a broader overview to this area, the reader is referred to [SS02a, SS02b] and [MG15b, MG15a] for stereo estimation techniques and to [Sei+06b, Sei+06a] and [Str+08b, Str+08a] for multi-view stereo reconstruction algorithms.

2.2.1 Shape from Silhouette

The reconstruction of 3D information from multiple image silhouettes of an object was first introduced by Baumgart [Bau74]. The principal concept of this approach is the approximation of a 3D object by the intersection of the back projections of the silhouettes' viewing cones. As a consequence, only the bounding geometry of an 3D object can be addressed. In order to emphasize this limitation, Laurentini introduced the term Visual Hull [Lau94].

Conceptually, there are three major flavors of Visual Hull algorithms. First, the *polyhedral approach*. The Visual Hull is computed by exploiting geometric properties and applying complex mesh intersection operations in order to receive a mesh representation, e.g. [FB09, FB03, MBM01, LFP07, Che+10, DR11]. Second, the *volumetric approach*. The volume of interest is voxelized and the voxels whose projections are not part of all silhouettes are discarded. The remaining voxels constitute the Visual Hull, e.g. [Soa+07, WTM06, LBN08, Pot87, NFA88, AV89, Sze93]. And third, the *image based approach*. As the works of section 3.1 are founded on this approach, a more detailed overview is provided in the following.

The Image Based Visual Hull approach allows for the direct computation of depth maps. An arbitrary desired image is defined and the viewing rays of individual pixels are intersected with the back projected cones of the silhouettes. The resulting 3D intervals are intersected and the depth of each individual pixel can be extracted from the closest 3D interval that is computed for this pixel position. An early work on the Image Based Visual Hull was reported from Buehler *et al.* [Bue+99]. Instead of a naïve computation of the line intersection for all pixel positions, the authors propose the application of a cache structure that caches the results for all pixel positions that lead to the same epipolar line and in consequence to the same interval intersections. Here, the silhouette intersections are computed by tracing the epipolar line through the image. For indexing the results within the cache structure, the farthest intersection with the image border is used. The required discretization for the binning of the results allows for performance scaling, but also introduces quantization artifacts. An extension of this approach that allows for an exact computation of the intersections was proposed by Matusik *et al.* [Mat+00, Mat+02b]. Instead of the image border, the edges of the silhouettes are used as a data dependent non equidistant index for the intersection cache. Here, the ordering within the cache structure depends on the scan-line direction in the desired image. In a follow-up work, Matusik *et al.* proposed a lazy evaluation strategy for line segment computation in order to improve the algorithmic performance [Mat+02a]. While the cache structure was switched back to the initially proposed variant [Bue+99], the line segments are extracted from the silhouette images on demand with a Bresenham algorithm. After each extraction, the cache is incrementally populated. An extension regarding quality and robustness was proposed by Grauman *et al.* [GSD03]. The authors apply a Bayesian framework to robustify the results against segmentation errors. In recent years, as massively parallel hardware architectures like modern graphics cards emerged, specialized algorithmic developments for an efficient and parallel IBVH processing on many-core systems were reported [Wai+09, Hau+12]. Here, the key algorithmic components for cache generation, 2D line intersection tests and 3D interval intersections are replaced with parallel counterparts.

2.2.2 Real-time Stereo

There is a huge amount of literature on stereo estimation algorithms. An extensive review is out of scope within this section. For an introductive overview, the reader is referred to [SS02a, SS02b]. In the following, the focus of this section is on a selection of algorithms that are either employed for video communication scenarios or known to work robustly and reliable in real-time production environments or have a conceptual relationship to the presented work.

Until today, many stereo algorithms incorporate basic functionalities from approaches that have been developed in the early years of stereo processing. The principles of the early work on the comparison of different spatial planes (*Plane-Sweeping*) [Col96] or image blocks (*Block-Matching*) [BF82] for the evaluation of local stereo hypotheses are still part of many approaches in the real-time stereo domain. In recent years, GPU-based Plane-Sweep approaches [Rog+09b, Rog+09a] have been applied to solve the eye contact problem [Dum+10, Dum+09, Dum+08]. Similarly, another Plane-Sweep variant that can include multiple views [Yan+04] has been applied within a medical telepresence scenario [Wel+05]. Conceptually, Plane-Sweeping has two points of contact with the Iterative Patch-Sweep (IPS) algorithm that is presented in section 3.2. First, both the spatial planes and the patches that are evaluated during the sweeping procedure represent the potential object orientation. However, while each sweeping plane determines a constant predefined object orientation across the whole image, the proposed IPS approach allows for an individual, pixel-wise surface representation. And second, both algorithms evaluate a list of hypotheses, i.e. 3D patches or 3D planes. But in contrast to Plane-Sweep, the list of hypotheses is not predefined with IPS. In case of rectified stereo configuration, the estimation of horizontal disparities has been pursued in many approaches. Based on the basic block-matching idea, different approaches for the definition and computation of *optimal* disparity values are proposed. For this task, the application of dynamic programming has been reported in order to balance between computational complexity and the incorporation of greater image regions instead of pure local matching, e.g. [Cri+03, LBW95] and [OLC93] that is founded on some early work from Cox *et al.* [Cox+92]. Other authors propose the application of hierarchical block-matching schemes implemented in hardware solutions [Ohm+98] or even a global optimization of the disparity map in terms of post-processing the results of an adaptive window-based matching approach [Tsa+04]. While the listed disparity estimation algorithms are diverse, all of them are designed to evaluate a fixed disparity range. Regarding this characteristic, the authors of [Div+10, KS02, LH02] propose a Hybrid Recursive Matching (HRM) algorithm that is not restricted to a predefined set of disparity values [Fel+10, Fel+09, KSO01]. Instead, the information is propagated via a meander-wise traversal of the image. New disparities are generated by disparity updates that are computed based on optical flow principles. While the meander-wise image traversal cannot be parallelized, regarding information propagation there is a conceptual relationship to the parallel, iterative and pixel-wise neighborhood propagation of the IPS algorithm. Additionally, IPS applies the same principle of temporal predecessor propagation as HRM. Based on the initial work on HRM, there has been some effort towards a parallelization on multi-core platforms. While the resulting Line-

Wise Hybrid Recursive Matching (L-HRM) algorithm [RZK11, Rie+12b] is employed in the domain of entertainment, broadcasting and post-processing like stereo to multi-view conversion [Rie+12a], the conceptual relationship to the presented approach remains since L-HRM conducts line-wise disparity information propagation.

In other real-time application domains like autonomous driving, robotics and areal photography, the Semi-Global Matching (SGM) approach [Hir05] has been intensively investigated and implemented for about a decade. While SGM is conceptually not related to the IPS algorithm, it comprises comparable properties regarding parallelizability, algorithmic performance and the applicability to real-time domains. Based on the initial work on SGM, there have been improvements in terms of untextured area handling [Hir08, Hir06] that resulted in the Consistent Semi-Global Matching (CSGM) algorithm. Other authors proposed an iterative Semi-Global Matching (iSGM) algorithm [HK12] with focus on driver assistance systems. Here, an algorithmic optimization is realized via the reduction of the disparity search space by an iterative cost path evaluation. At the same time, a parallel implementation on graphics hardware has been provided [EH08] and further improved [Mic+13] in order to enable SGM for the application to real-time domains. As graphics hardware is not always available in application domains with embedded architectures, there have been additional developments for the implementation of SGM on FPGAs [HBE12, HOP14, Bud12, Ban+10]. In the automotive domain, specialized algorithms for processing road surfaces have been developed [EE13, GSF14]. Here, the prior knowledge about the fixed type of stereo scene is used to guide the algorithm while performing the 3D analysis. The authors propose either the preprocessing of the input images in order to improve the matching process [EE13] or to create a mean disparity map for the average road surface that can be used as a cue for disparity selection [GSF14].

From a conceptual point of view, together with the already discussed HRM and L-HRM algorithms, PatchMatch Stereo [BRR11] comprises a close relationship to IPS. Based on an algorithm for patch-based image editing [Bar+09], the basic idea has been extended to stereo processing. Here, the conceptual overlappings with the IPS algorithm are as follows: The initial patch parameters are randomly drawn for both algorithms. The meander-wise image traversal for spatial information propagation that is used in HRM, is also part of the iteration step of PatchMatch Stereo. Consequently, as with HRM, there is a conceptual link to the pixel-wise iterative information propagation of IPS. Moreover, HRM and PatchMatch Stereo also share the same principle of temporal predecessor propagation that has been discussed earlier while comparing HRM and IPS. Finally, PatchMatch Stereo introduces a randomized plane refinement step that serves for the same purpose as the proposed Monte Carlo based hypotheses update of the IPS. While various enhancements [Hei+13, Lu+13, Bes+14, Xu+15] for PatchMatch Stereo have been proposed, no conceptual principles were added that can be considered as being related to IPS. In [Hei+13], a Huber regularized variational smoothing has been proposed that is applied after each PatchMatch iteration. Here, the optimization is carried out on a relaxed version of the formulated energy term via a primal dual formulation of the Huber-Rudin-Osher-Fatemi (Huber-ROF) model. While the stereo results could be significantly improved in comparison to the initial PatchMatch

algorithm, the runtime has been lifted to the range of minutes. The authors of [Lu+13] included an edge-aware filter to the randomized search and use a superpixel representation for the matching procedure. The runtime of the improved algorithm is reported to be in the range of seconds for the Middlebury benchmark dataset [SS02b]. An extension with focus on the optimization of a global data term via belief propagation has been proposed in [Bes+14]. Among the discussed PatchMatch extensions it is the computationally most expensive variant with more than 1000 seconds processing time for 0.3 megapixels [Xu+15]. Another variational approach based on the Potts model has been proposed in [Xu+15]. Beside stereo processing, this specific extension enables for a joint object segmentation and 3D analysis while offering a runtime of several hundred seconds on moderate image sizes.

2.3 Photometric Alignment

Various applications including aerial photography, image mosaicing, multi-view and stereo processing or view synthesis from multiple images benefit from photometric alignment. Here, the huge amount of different application targets makes an exhaustive benchmarking of photometric alignment algorithms difficult. However, there are individual works on the evaluation and comparison of algorithms in the domain of multi-view video coding [FL15b] and for multi-view image and video stitching [XM10]. As a colorimetric adjustment by definition involves the transformation of the image colors, the reader is referred to [Far+14] for a comprehensive survey on possible color mapping function and to [GN03] for a theoretical discussion on the estimation of intensity mappings. While there are generic approaches for the colorimetric normalization of individual images without any coupling among views such as [FSC98], many dedicated approaches were reported. The authors of [KP08, KFP08], for instance, propose a specialized algorithm for the estimation of the radiometric response function and the exposure of a camera in order to compensate for illumination changes in static outdoor scenes.

In compliance with the algorithmic developments of this thesis, in the following, the focus will be on algorithms for the photometric alignment in the domain of stereo and multi-view 3D processing. Beside the technical facilitations for 3D estimation [HS07], color correction is an important precondition for a convenient perception of rendering results [Pöl+12]. In most works, a photometric alignment is performed on the basis of an underlying (sparse) geometric image registration. Here, the image correspondences are used to identify image regions that are expected to exhibit identical colorimetric properties. A frequently applied approach for the matching of image regions is the detection of a colored calibration pattern. Based on a planar color-coded checkerboard pattern, Ilie *et al.* [IW05] propose a photometric inter-camera calibration with an algorithmic pipeline for the update of camera hardware parameters and a subsequent software-based color refinement with a linear or polynomial RGB to RGB mapping function. While pursuing a similar approach, the authors of [LDX10, LDX11] use an omni-directional color checker in order to simplify the registration of widely distributed cameras. On the basis of a colored omni-directional pattern, a more integrated framework that includes camera calibration, adjustment of camera hardware settings and a

subsequent color alignment among cameras is presented in [Ket+10]. Targeting on multi-view panoramic cameras, an even more integrated checkerboard approach that additionally includes the global optimization for panorama stitching and multi-view epipolar alignment is proposed by Kurillo *et al.* [Kur+13].

Pattern-based approaches can provide reliable correspondences among images and the outcome of the photometric registration mainly depends on the choice of the color mapping. However, the requirement for color charts render these kind of algorithms into semi-automatic procedures. In order to avoid manual interaction, feature point based image correspondences have been exploited. For this purpose, in [YO08] an energy minimization is used to optimize a lookup table for a color mapping between matching image regions that are identified via SIFT features. Similarly, the authors of [Pan+10] use SIFT features in conjunction with a dynamic programming approach for the minimization of a photometric energy term. An optical flow-based variant of SIFT for the identification of corresponding image regions is proposed by Wang *et al.* [WSW10]. In contrast to many other approaches, the color correction is performed individually for each image region on the basis of a custom color mapping in HSI color space. Beside SIFT based approaches, there are other works that rely on SURF features [FL15a] or propose their own custom approaches for correspondence mapping such as a region-wise histogram matching [Lu+15]. Instead of feature point matching, other authors apply (sparse) stereo for the identification of color correspondences. In this context, Doutre *et al.* [DN09] introduced a least squares regression for the optimization of a color mapping function with respect to average colors of image regions that were identified via sparse block-matching results. A more advanced matching via the extraction of 3D patches among images that is conducted similar to [FP07] is proposed in [NKM10]. As sparse image correspondences in general can only reflect a subset of the variety of colorimetric differences between images, Baskurt *et al.* [Yve+14] proposed a global color matching via polynomial regression for color correspondences that are computed based on a dense disparity estimation. While also aiming on an image wide color matching, the authors of [WFE11] propose an alternating iterative refinement of disparities and the color matching result. The formulation of the iteration procedure includes a globally optimal estimation of the disparity field with respect to the current color settings and the application of an arbitrary user-defined color mapping. Concentrating on a single video camera, the authors of [HE09] propose an extension of the optical flow equation in order to simultaneously estimate the geometric deformation of a predefined deformable surface mesh and the temporal changes of the photometric camera parameter. The proposed approach focuses on the real-time rendering of a new texture onto a moving and deforming surface in the original video. However, in some cases sufficient image correspondences might be expressed parametrically. In case of planar scenes for instance, a homography mapping between images can be applied. The geometric registration is reduced to the estimation of the eight homography parameters, e.g. [Bar08].

2.4 Calibration of the Eye Contact View

As presented in section 2.1, a comprehensive amount of work was conducted regarding eye contact preserving video communication. In contrast, procedures for the positioning of the eye contact camera received only little attention. On the one hand, pure teleprompter-like techniques with half transparent screens as reviewed in section 2.1.4 allow only for fixed predefined positions for the *eye contact camera*. In consequence, there is no freedom to algorithmically adjust the eye contact view. On the other hand, despite tackling the eye contact problem with image processing approaches, many authors pursue a manual placement of the *eye contact camera* in their work [Dum+08, Dum+10, Cri+03, Gem+00, GZ02, Ver+03, JD02]. This leads to a similar situation as with a predefined and fixed *eye contact camera*. Hence, eye contact is only provided if the conferees remain at the resulting *sweet spots* during conversation. In case of moving conferees, a fixed *eye contact camera* only provides an approximate eye contact view.

While many authors disregard the development of procedures for the placement of the *eye contact camera*, there is one mentionable exception. Kuster *et al.* [Kus+12b] propose two mechanisms for eye contact calibration. The first one relies on a manual user interaction based on a trackball-like interface for choosing the eye contact perspective. The second one is a semi-automatic approach, were the user is captured with a Kinect sensor in two different positions. The transformation information that is extracted from these positions is subsequently used to compute the *eye contact camera*. However, both approaches still result in a fixed camera configuration and therefore restrict eye contact to the *sweet spot* positions.

3. Real-time Multi-view 3D Analysis

The ultimate goal of this thesis is the generation of a synthesized view that enables for the provision of eye contact. As discussed in section 2.1.1, for this purpose, 3D information of the conferee is required in order to generate a perspectively correct result. Depending on the application domain and the completeness and quality of the available 3D data, many different approaches for view synthesis were reported, e.g. [Rie+12a, Mur+10, Mül+09, Yan+04, Coo+03]. However, due to specializations for limited 3D information, in general there are restrictions with these methods regarding the possible viewing perspectives. The work of Riechert *et al.* [Rie+12a], for example, is limited to a perspective that is located on the baseline between two real cameras. In contrast, if sufficiently complete, high quality 3D data is available, the virtual camera can be moved freely and the novel viewing perspective can even be directly rendered via a 3D graphics API like OpenGL. As it is discussed in chapter 5, flexibility regarding the position of the virtual camera improves the naturalness of the conversation, but also enhances the user experience as there is no single *sweet spot* where the conferee needs to be located in order to establish eye contact. Recent algorithms in the domain of multi-view stereo are able to produce very complete and high quality results with respect to laser scanned references [Sei+06b]. In section 3.2.4.5, it is exemplarily shown based on the work of Furukawa *et al.* [FP10], that multi-view stereo output is well suited for video communication application to render virtual eye contact views. However, the works on multi-view stereo focus mainly on the reconstruction quality and not on computing time. The reported runtime for a single frame is commonly in the range of minutes or even hours [Sei+06a].

In this work, the multi-view 3D analysis is performed on basis of multiple stereo or trifocal camera systems that are distributed around the volume of interest. Criteria like the size of the communication screen and its resolution, operating range of the conferees and the desired circumference of possible rendering perspectives for synthetic views determine the number of required cameras and their adequate positions and orientations. In order to capture a large portion of the conferees' geometry, in this thesis a camera setup with sixteen cameras was used to render the eye contact view, c.f. appendix A and B. While the cameras were always grouped as stereo or trifocal systems, these camera pairs or triples were widely distributed across the scene in order to provide many different perspectives. The resulting camera configuration is algorithmically approached via two complementary 3D reconstruction techniques. The narrow stereo or trifocal groups are used for a patch-based reconstruction of the object surface, while the wide distribution of cameras is used for an

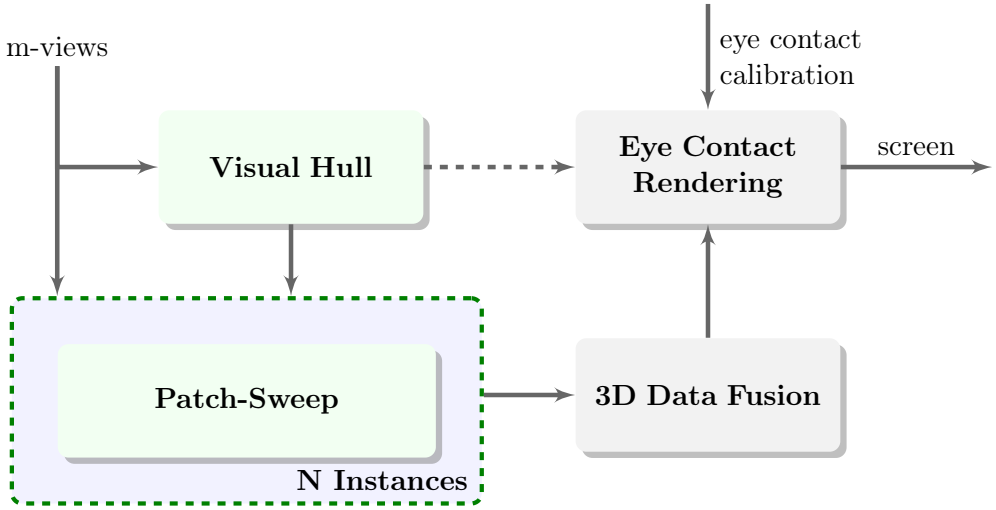


Figure 3.1: Algorithmic structure of the multi-view 3D analysis and eye contact view synthesis with N stereo and/or trifocal camera systems.

approximation of the object via its Visual Hull. In this context, it is assumed that the background is already removed by a state-of-the-art foreground segmentation algorithm such as [CE10]. Both reconstruction approaches can be applied individually or in combination as it is discussed in section 3.2.4.4. The main purpose for this combination is the suppression of stereo mismatches via shape-based depth data. Especially in unstructured and homogeneous image regions, many mismatches occur due to the multitude of ambiguities. For a challenging example please refer to the `Sylvain` dataset in appendix B. While other works use commodity depth sensors such as the Microsoft Kinect for problematic image regions or as a stand-alone solution [MF12, Mai+12], there is a downside compared to the proposed approach. In general, consumer depth sensors interfere if more than one device is used and the resulting depth output degrades. In addition, the consumer depth cameras that are on the market so far cannot be synchronized among each other or with other cameras. The potential time shift intricates the camera calibration procedure and might cause temporal artifact. Alongside a robust and precise multi-view 3D reconstruction, there are application-inherent real-time constraints and additional demands regarding the temporal consistency of the eye contact corrected video stream. In this work, the real-time constraints are addressed by an algorithmic design that focuses on massive parallel processing with modern graphics cards, and the temporal consistency of 3D data is implicitly maintained via the inclusion of temporal predecessors for the patch-based reconstruction. The algorithmic structure for the proposed multi-view 3D processing and eye contact synthesis is illustrated in figure 3.1. Comparable to [BBH08], individual stereo results are combined into a joint 3D model. For this task, all stereo results are transformed into point clouds and fused to patch groups based on geometric plausibility and visibility constraints. Here, a state-of-the-art GPU based real-time point cloud fusion is used [Ebe+14]. Due to the comprehensive amount of 3D data, the fusion is able to eliminate many mismatches and the final results are comparable to other state-of-the-art multi-view approaches that exhibit a much higher computation time.

The main contributions of this chapter are a novel variant of an Image Based Visual Hull

(IBVH) algorithm and a novel algorithmic framework for patch-based stereo estimation. This framework is capable of including IBVH depth data as hypotheses seeds in order to improve the completeness of the results especially in case of unstructured homogeneous image regions. The proposed IBVH variant is based on existing work of Matusik *et al.* [Bue+99, Mat+00, Mat01, Mat+02b, Mat+02a]. The focus of the enhancements was on a massively parallel algorithmic design that allows for the real-time computation of high resolution depth maps on parallel architectures like graphics hardware. As with the original work on IBVH, the depth map computation can be based on an arbitrary desired camera. However, the key components for the identification of potential line segment intersections and the required tests for 2D and 3D line intersections were replaced by efficient parallel counterparts. Additionally, an implicit extension of the object is integrated for cases where the silhouette of the object is restricted by the image border. The IBVH results can be used individually or in combination with the novel patch-based real-time stereo algorithm that is denoted as Patch-Sweep. It is based on the iterative evaluation and propagation of 3D hypotheses in terms of 3D patches. The parallel design of the algorithm allows for an efficient implementation on graphics hardware in order to maintain real-time constraints. As the hypotheses evaluation is based on a temporal interdependency with the previous frame, the algorithm exhibits a high rate of convergence for a consecutive input sequence. But even when starting from an uninitialized state, in general a converged state is reached after one or two iterations. In consequence, the algorithm enables for an efficient real-time computation of multiple stereo input streams. Additionally, the hypotheses-based algorithmic structure allows the inclusion of shape-based depth information from IBVH or any other source of 3D data. The proposed combination of Patch-Sweep and IBVH mitigates the impact of mismatches for homogeneous image regions and constitutes a convenient alternative to the integration of depth sensors.

This chapter consists of two sections. In section 3.1, an inherently parallel real-time and high resolution Visual Hull algorithm is presented. The Patch-Sweep algorithm that is designed for real-time 3D in case of narrow baseline stereo and trifocal camera configurations is introduced in section 3.2. As illustrated in the structural overview in figure 3.1, both techniques can be combined for real-time multi-view 3D analysis. Parts of this chapter have already been published in [WFS11, Wai+11, Wai+13, Wai+09, 13a, Ebe+14].

3.1 High Resolution Depth Maps from Visual Hull

The development of a real-time high resolution Visual Hull (VH) for depth map computation is motivated by two objectives. First, in case of widely distributed cameras and moderate correction of the viewing perspective, the VH results can directly serve as an input for virtual eye contact rendering. Second, VH can be used in combination with the Patch-Sweep algorithm as discussed in section 3.2.4.4. Here, the VH provides an initial cue for the patch-based 3D processing of the input images. In general, any flavor of Visual Hulls could be used for depth map computation as the result can always be transformed to depth information. However, in this work, the focus is on the Image Based Visual Hull (IBVH)

that conceptually supports the direct computation of depth maps. For this purpose, an arbitrary reference view is defined. This view does not need to be one of the input views. For each pixel of the reference view, its corresponding viewing ray is intersected with all back projections of the object silhouettes from the input views. In the following, these back projections will be denoted as generalized cones. The closest intersection point constitutes the depth value for the pixel position of the intersected viewing ray.

The main idea of this section is to algorithmically extend existing work on IBVH [Bue+99, Mat+00, Mat01, Mat+02b, Mat+02a] in order to enable for real-time processing of high resolution input on parallel architectures like graphics hardware. In this context, the major contributions of this thesis are a novel cache structure for the efficient parallel identification of potential line segment intersections, an integrated image border extension that is applied in cases where an object is only partially visible in some views and the parallel handling of the 2D and 3D intersection tests. The proposed cache structure allows for a computationally cheap preselection of pixel coordinates of the reference view that are relevant for depth map computation. The preselected pixel coordinates constitute a close boundary to the image area where the object is located. In contrast to a complete processing of all pixels of the reference view, the restriction to the preselected area allows for a significant speedup.

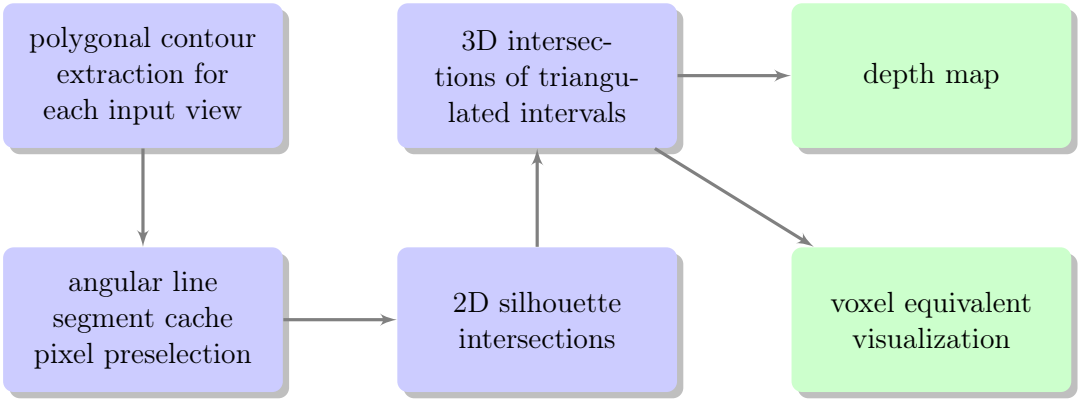


Figure 3.2: Structure of algorithmic components and data flow of the proposed parallel VH algorithm.

The four components of the presented algorithm are illustrated in figure 3.2. The algorithmic design of each component is focused on parallel processing. For each input view, a polygonal contour of the silhouette is extracted. The resulting line segments are assigned to the angular line segment cache of the respective view, and a preselection of pixels for the reference view is carried out based on the extremal angular values of the cache entries. The intersections in image space are efficiently computed by cache-aided intersection tests. For each silhouette image, all intercept points are lifted to 3D. Finally, the VH is created by the intersection of the emerging 3D intervals. Without any algorithmic overhead, these intervals can be used for depth map extraction or for the rendering of a voxel equivalent visualization of the 3D object. In the following, the extraction of polygonal contours and the representation of the extracted line segments for visual hull computation is described in section 3.1.1. Afterwards, the parallel computation of the angular line segment cache that is

needed for intersection tests and the preselection of pixels of the reference view is discussed in section 3.1.2. Pixel preselection, the subsequent 2D line intersection together with the 3D interval computation, and the extraction of depth data are addressed in section 3.1.3. Additionally, a close relationship to voxel based methods that allows for a voxel equivalent visualization is covered in this section.

3.1.1 Polygonal Contours and Line Segment Representation

For each input view, the silhouette information is transformed from pixel representation to a list of line segments that constitute a polygonal contour of the foreground pixels. Algorithmically, this polygonal representation of the silhouette border is extracted by means of marching squares, the 2D version of the well-known marching cubes algorithm [LC87]. By design, the line segments extracted with marching squares span only from one pixel to a neighboring pixel. While these line segments could be potentially combined with no or little loss in the precision of the contour, in this work the untouched output of marching squares is used as input for further processing. Regarding a massively parallel implementation on graphics hardware, this decision is motivated from an engineering point of view. For the amount of line segments that is extracted in the current application domain, a combination of line segments is more time consuming than the additional execution costs for a greater amount of line segments. In order to address the partial visibility of objects, a distinction between two cases is required during line segment extraction. There is the regular case, where the silhouette border is within the image and the partial visibility case, where the image border constitutes the silhouette border. In the second case, the line segment is marked with a border flag in order to enable for interval intersections of infinite length as discussed in section 3.1.3 in more detail.

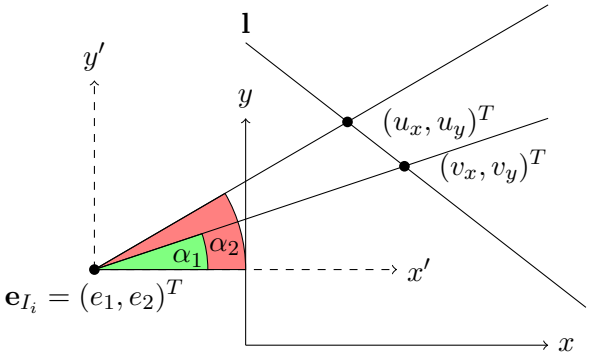


Figure 3.3: Line segment representation with interval restricting angular values. Marching squares identified a line segment from $(u_x, u_y)^T$ to $(v_x, v_y)^T$.

The line segments of the polygonal representation of the silhouette borders are organized in a cache structure. The purpose of this cache is to easily identify the line segments that are subject to intersection tests for a given viewing ray. Additionally, in advance to any intersection test, the cache allows for a computationally cheap preselection of relevant pixels in the reference view. The cache organization is based on an angular relationship between the line segments and the silhouette image epipoles, i.e. the epipoles of the reference

image and the input views. Within the cache structure, a line segment is represented as a tuple $\mathbf{L} = (\mathbf{l}, \boldsymbol{\alpha})$ where $\mathbf{l} \sim (l_1, l_2, l_3)$ denominates a line in homogeneous coordinates and $\boldsymbol{\alpha} = (\alpha_1, \alpha_2)$, $\alpha_1 < \alpha_2$ are angular values that restrict this line to an interval. Let I_E be the reference image, $\{I_i\}_{i=1}^m$ the set of silhouette images, $\mathbf{e}_{I_i} = (e_1, e_2)^T$ the epipole of the image pair I_E, I_i in image I_i and $\mathbf{u} = (u_x, u_y, 1)^T$, $\mathbf{v} = (v_x, v_y, 1)^T$ the homogeneous representation of the current line segments start and end point as extracted by the marching squares algorithm. The resulting line that connects both points can be directly computed as the cross product $\mathbf{l}^T \sim \mathbf{u} \times \mathbf{v}$. In compliance with figure 3.3, the computation of the line restricting angular values $\boldsymbol{\alpha}$ reads as

$$\boldsymbol{\alpha} = (\min(\beta_1, \beta_2), \max(\beta_1, \beta_2)), \tag{3.1}$$

where $\beta_1 = \arctan2(v_y - e_2, v_x - e_1)$ and $\beta_2 = \arctan2(u_y - e_2, u_x - e_1)$. In order to provide a compact angular range of $[0, \ldots, \pi]$, the original $\mathbf{L}$-tuples are mapped according to their $\boldsymbol{\alpha}$ values as

$$(\mathbf{l}, \alpha_1, \alpha_2) \leftarrow \begin{cases} (\mathbf{l}, \alpha_1, \alpha_2) & \alpha_1 \geq 0 \text{ and } \alpha_2 > 0 \\ (\mathbf{l}, \alpha_1 + \pi, \alpha_2 + \pi) & \alpha_1 < 0 \text{ and } \alpha_2 \leq 0 \\ \{(\mathbf{l}, 0, \min(\alpha_1 + \pi, \alpha_2)), (\mathbf{l}, \max(\alpha_1 + \pi, \alpha_2), \pi)\} & \text{sign}(\alpha_1) \neq \text{sign}(\alpha_2), \end{cases} \tag{3.2}$$

where the last row of the case distinction denotes a split into two individual subsegments. In this way, all line segments that might be subject to an intersection test with a certain epipolar line can be identified via the requirement that their angular ranges need to include the angular value of the epipolar line, c.f. figure 3.4. From a theoretical perspective, this representation is only applicable in case the epipole is not located at infinity. However, the restriction is not serious since the targeted practical camera setups do not comprise this configuration.

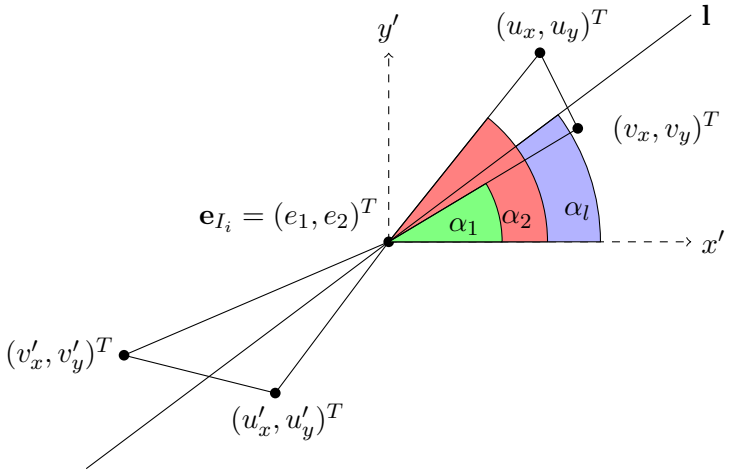


Figure 3.4: Line segment intersection test based on angular values. Each line $\mathbf{l}$ always intersects an even number of line segments. Here, the line segments $\left((u_x, u_y)^T, (v_x, v_y)^T\right)$ and $\left((u'_x, u'_y)^T, (v'_x, v'_y)^T\right)$ have been intersected.

3.1.2 Angular Line Segment Cache

In the following, the functionality of the cache is explained in three steps. First, the layout of the cache bins is described. Second, the mapping of the line segments to cache bins is introduced and third, it is explained how a viewing ray is mapped to the cache bin that contains the potentially intersecting line segments. For each silhouette image a separate cache structure with N_B bins is built. The number of cache bins is directly linked to the computational efficiency. In order to identify the optimal number, the impact of a varying cache bin count was empirically evaluated in section 3.1.4. Each cache bin contains line segments for a certain angular range, where the total maximal angular range of the cache is defined as $[0, \ldots, \pi]$. In practice, the required angular range for a certain image is smaller. In consequence, the angular range is selected in order to cover the effective range of angular values of the respective line segments. A mapping from the modified $\boldsymbol{\alpha}$ values of the line segments as defined in equation (3.2) onto discrete cache bins is given in the following. Let $\alpha^I_{\min}$ and $\alpha^I_{\max}$ the minimum and maximum of the modified set of α values for image I as exemplarily illustrated in figure 3.5. Then, the angular value assigned to bin $b^I_j, j \in \{0, \ldots, N_B - 1\}$ is defined as $\alpha^I_j := j \cdot \frac{\alpha^I_{\max} - \alpha^I_{\min}}{N_B - 1} + \alpha^I_{\min}$. Accordingly, as shown in figure 3.6, depending on its angular range, each line segment is attached to all bins b^I_j with $j_{\min} \leq j \leq j_{\max}$, where

$$j_{\min} = \left\lfloor \frac{(N_B - 1) \cdot (\alpha^I_1 - \alpha^I_{\min})}{\alpha^I_{\max} - \alpha^I_{\min}} \right\rfloor \text{ and } j_{\max} = \left\lfloor \frac{(N_B - 1) \cdot (\alpha^I_2 - \alpha^I_{\min})}{\alpha^I_{\max} - \alpha^I_{\min}} \right\rfloor. \tag{3.3}$$

The intersection of a viewing ray from the reference image with the generalized cone is computed in image space. The projection of the viewing ray onto the silhouette image constitutes the epipolar line $\mathbf{l} \sim (l_1, l_2, l_3)$ that corresponds to the coordinates of the viewing ray in the reference image. As the epipole is in line with $\mathbf{l}$, the angle for intersection testing with respect to the mapping equation (3.2) reads as $\alpha_l = \arctan(-l_1/l_2) + \frac{1-\text{sign}(-l_1/l_2)}{2}\pi$. Consequently, $\mathbf{l}$ intersects all mapped line segments with angular values that fulfill $\alpha_1 \leq \alpha_l \leq \alpha_2$ as illustrated in figure 3.4. In order to look up potentially intersecting line segments, the cache index i of the bin that contains these segments can be identified as $i = \left\lfloor \frac{(N_B - 1) \cdot (\alpha_l - \alpha_{\min})}{\alpha_{\max} - \alpha_{\min}} \right\rfloor$.

3.1.3 Pixel Preselection, 3D Interval Computation and Intersection

The computation of 3D intervals that emerge through the intersection of viewing rays of the reference image with all generalized cones and the intersection of these intervals is divided into three steps. First, in order to speed up the subsequent intersection tests, the pixels positions of the reference image that might contain valid object intervals are preselected according to the extremal angular values of the silhouette views. Second, the intersection of each viewing ray with every generalized cone is computed in image space via the intersection of the corresponding epipolar lines and the polygonal contours. And third, the 2D intersection intervals are triangulated in order to obtain the final result by intersecting the resulting 3D intervals. Each of these steps can be done in parallel on graphics hardware for all pixels of the reference image.

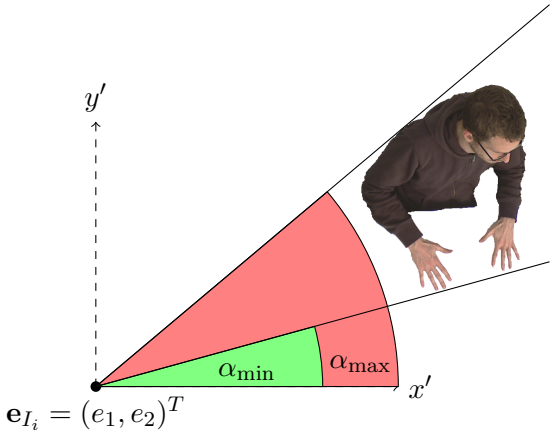


Figure 3.5: Pixel preselection according to minimal and maximal angular cache entries $\alpha_{\min}$ and $\alpha_{\max}$. Pixel coordinates not satisfying the angular constraint can be excluded from subsequent computations.

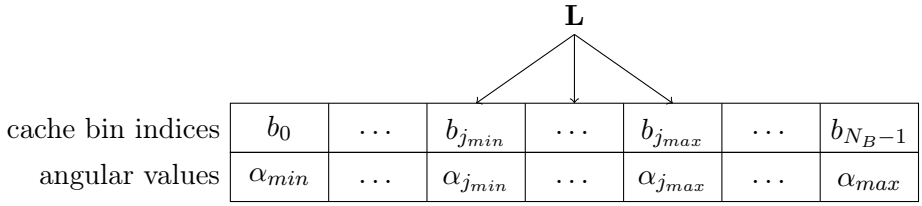


Figure 3.6: The angular values assigned to the cache bins are quantized with respect to the extremal angular values of the silhouette mask α_{min} and α_{max}. According to equation (3.3), the line segment $\mathbf{L}$ is assigned to every cache bin b_j, $j_{min} \leq j \leq j_{max}$ within its angular range.

For all pixel coordinates on the image plane of the reference image $\mathbf{x}_E \in \Omega_E$, a computationally inexpensive preselection based on the extremal angular values stored in the cache structures of all silhouette images is done by an epipolar transfer of the pixel coordinates. The angular value in image I_i that corresponds to the viewing ray for the pixel position $\mathbf{x}_E$ is given by $\alpha^{I_i}_{\mathbf{x}_E} = \arctan(-l_1/l_2)$, where $(l_1, l_2, l_3) \sim \mathbf{F}_{I_iE}\,\mathbf{x}_E$ and $\mathbf{F}_{I_iE}$ denotes the fundamental matrix that maps points from the reference image to lines in the silhouette image I_i. In case $\alpha^{I_i}_{\mathbf{x}_E} < 0$, the mapping $\alpha^{I_i}_{\mathbf{x}_E} \leftarrow \alpha^{I_i}_{\mathbf{x}_E} + \pi$ is needed in order to compare this angular value with the extremal cache entries. As illustrated in figure 3.5, pixel coordinates of the reference image need only to be considered for subsequent computations if they fulfill the angular condition $\alpha^{I_i}_{\min} \leq \alpha^{I_i}_{\mathbf{x}_E} \leq \alpha^{I_i}_{\max}$ for all silhouette images I_i. The resulting set of preselected pixels can be interpreted as the projection of a non-cuboidal bounding volume of the 3D object. Due to this approximation of the final silhouette in the reference image, a significant algorithmic speedup is constituted. The 2D silhouette intersections are computed based on an epipolar transfer of each preselected pixel onto every silhouette image and a cache lookup according to section 3.1.2. The intersection coordinates are obtained by determining the intercept points of the epipolar line and the line segments retrieved from the cache. In order to enable for the handling of partially visible objects, potential border flags that are attached to the intersected line segments are transferred to the respective intersection coordinates. Then, for each pixel coordinate of the reference image, the 3D

intervals for the 2D intersection results are computed. However, depending on the camera configuration and the size and position of the object of interest, there are degenerated configurations that might cause some triangulation results to be located at the back of the reference image. An illustration of such a situation is provided in figure 3.7. The angle between the current viewing ray and the generalized cone causes an illegal result for the interval point $\mathbf{X}_b$. Since no other information about $\mathbf{X}_b$ is available, in this case the 3D interval is set from $\mathbf{X}_a$ along the viewing ray until infinity. Subsequently, information about partial object visibility is transferred to the 3D intervals. If the originating 2D coordinates exhibit a border flag, the start point and/or the end point of a 3D interval are expanded to 0 and ∞ respectively. In this way, the unknown extent of an object in image space is reflected in 3D. For the intersection of all computed 3D intervals of one preselected pixel, the intersection operations can be reduced to an algorithmically efficient comparison of the coordinates z-values if the camera coordinate system of the reference image is chosen to be aligned with the world coordinate system. Finally, the depth map of the reference view

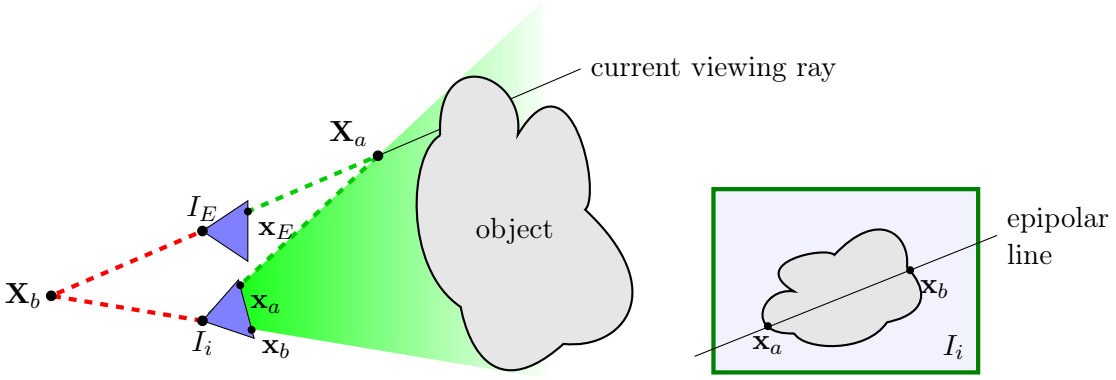


Figure 3.7: In this configuration, the second interval point $\mathbf{X}_b$ is triangulated behind the reference camera. **Left:** Illustration of the triangulated points in the 3D scene. **Right:** Polygonal contour intersection in image I_i.

is generated by collecting the smallest z-values of all interval sets and insert them to the corresponding positions. Alternatively, if a visualization of the VH is required, the set of 3D intervals can be rendered as 3D point cloud or small frustums for each interval in case a closed surface is needed.

For a voxel equivalent visualization, an orthographic projection, i.e. a reference view at infinity, can be used as follows. If the camera of the reference view is placed at infinity, the frustum shape of 3D intervals turns into a rectangular cuboid that can be used for a voxel equivalent representation as illustrated in figure 3.8. An appropriate camera at infinity can be obtained based on the axis-aligned bounding box of the volume that can be observed by the reference cameras. Let $x_{\min}$, $y_{\min}$, $z_{\min}$ be the minimal coordinate values of the bounding box for the three spatial directions and $x_{\max}$, $y_{\max}$, $z_{\max}$ the maximal coordinate values. The greatest desired voxel resolution M leads to the edge length of a single voxel

$$v_s = \frac{\max(x_{\max} - x_{\min}, y_{\max} - y_{\min}, z_{\max} - z_{\min})}{M}. \tag{3.4}$$

Consequently, the reference camera at infinity reads as

$$\mathbf{P}_{\text{inf}} \sim \begin{pmatrix} \frac{1}{v_s} & 0 & 0 & -\frac{x_{\min}}{v_s} \\ 0 & \frac{1}{v_s} & 0 & -\frac{y_{\min}}{v_s} \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{3.5}$$

In case of a cubic voxel shape, the computation of the voxel resolution (M_x, M_y, M_z) for the bounding box is straightforward and the resolution of the reference image can be set to $M_x \times M_y$.

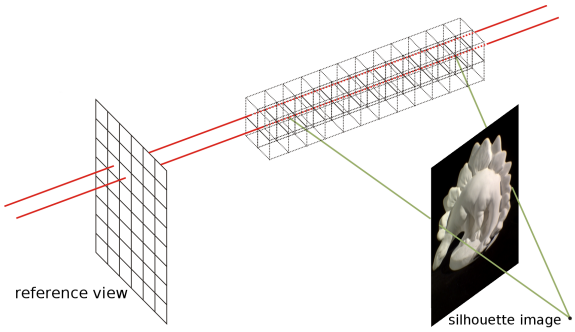


Figure 3.8: A voxel equivalent visualization is generated with a reference view at infinity. The computed 3D intervals on the parallel viewing rays of the camera at infinity are equivalent to a voxel based result.

3.1.4 Experiments

In the following, two aspects of the presented VH algorithm are evaluated. On the one hand, the results for depth map computation and voxel equivalent visualization are illustrated. On the other hand, the algorithmic efficiency for different input and output resolutions and the impact of pixel preselection and the selected cache size N_B is measured. All computations were conducted on a single NVIDIA Titan X graphics card. The experiments are based on the `Niklas`, `David`, `Sylvain`, `Marcus`, `Paul` and `Oliver2` datasets. Exemplarily, results for the `Niklas` dataset from appendix B are visualized. In figure 3.9, the 16 foreground segmented images together with a computed depth map for the reference view are illustrated. The selected reference view was not part of the input views, but was chosen as a new camera that is located in front of the person. The gray shaded area around the depth map indicates the preselected image area that was computed as described in section 3.1.3. It can be seen that this area already constitutes a close approximation to the object area. In figure 3.10, the voxel equivalent representation of the depth result from figure 3.9 is shown. The presented visualization exhibits no perceivable difference compared to a voxel based visualization. In order to demonstrate the impact of the interval extension for partially visible objects, the first row of figure 3.10 shows results with enabled border extension, while the second row shows the same results without border awareness. The red colored areas indicate the cutoff regions that are caused by border agnostic processing. In this context, please note that the cutoff at the belt line was caused due to the segmentation of the table that was in front of the person and is not related to any image border that restricts the objects boundaries.

For the evaluation of the algorithmic efficiency with respect to a varying cache bin count and the impact of pixel preselection, computation times for three different resolutions are measured. The input datasets and the reference view were scaled to High-definition (HD, 1920×1080), quarter HD (QHD, 960×540), and quarter QHD (QQHD, 480×270) resolution. First, the optimal number of cache bins was empirically identified. While the results are very similar throughout all datasets, the mean computation times for different N_B values are exemplarily plotted in figure 3.11 for the `Niklas` dataset. It can be seen that a good choice for N_B is within the range of $\left[2^{11}, \ldots, 2^{13}\right]$. For the following performance evaluations on runtime and on the impact of pixel preselection, the number of cache bins was set to $N_B = 2048$.

The comparison between enabled and disabled pixel preselection is illustrated based on a frame wise evaluation of the runtime on the `Niklas` dataset and on average values for all six datasets. The frame wise results are plotted in figure 3.12, while table 3.1 exhibits the average compute times. It can be seen that the pixel preselection leads to a significant speedup within all experiments. Here, the smaller speedups for the lower resolutions are mainly caused due to an underutilization of the graphics hardware. Table 3.1 also shows significant runtime differences between the datasets. From a content perspective, there are two main differences between the datasets. First, there are different movements of the persons and second, there is a different quality of foreground background segmentation. The datasets exhibit a different amount of background pixels that were not segmented properly. In consequence, the required interval intersections vary between the datasets. A greater number of interval intersections naturally accounts for a higher computational complexity.

Finally, the quantization effects on QHD and QQHD resolutions were evaluated. Therefor, the depth results for HD resolution were considered as a reference. In order to enable for the comparison of different results, the QHD and QQHD depth maps were upscaled to HD resolution based on nearest neighbor interpolation. The mean absolute depth differences in millimeter are listed in table 3.2 for all six datasets. While the number of pixels at HD resolution is four times higher compared to QHD and sixteen times higher compared to QQHD, the depth differences remain moderate. With a mean depth difference of less than 4 mm, even the QQHD resolution results could still serve as a good depth cue for subsequent processing as it is discussed in section 3.2.4.4.

	QQHD			QHD			HD		
	w/o pre	pre	su	w/o pre	pre	su	w/o pre	pre	su
`Niklas`	7.38	6.20	1.19	24.66	18.03	1.37	91.75	63.27	1.45
`David`	8.63	7.18	1.20	25.99	20.20	1.29	104.81	74.25	1.41
`Sylvain`	9.12	7.77	1.17	28.91	23.98	1.21	120.12	92.44	1.30
`Marcus`	9.82	8.43	1.17	29.75	26.23	1.13	125.61	98.94	1.27
`Paul`	9.41	8.39	1.12	25.75	22.98	1.12	106.98	86.71	1.23
`Oliver2`	9.51	7.83	1.21	31.69	25.47	1.24	123.18	90.57	1.36

Table 3.1: Average compute times in milliseconds without preselection (w/o pre) and with preselection (pre) together with the respective speedups (su). Three different resolutions have been processed.

	QHD vs. HD	QQHD vs. HD
`Niklas`	0.968 mm	1.712 mm
`David`	2.248 mm	3.851 mm
`Sylvain`	2.232 mm	3.719 mm
`Marcus`	1.459 mm	2.669 mm
`Paul`	1.968 mm	3.617 mm
`Oliver2`	1.852 mm	3.060 mm

Table 3.2: Mean absolute depth differences with respect to the HD results. For each listed dataset 100 frames were processed.

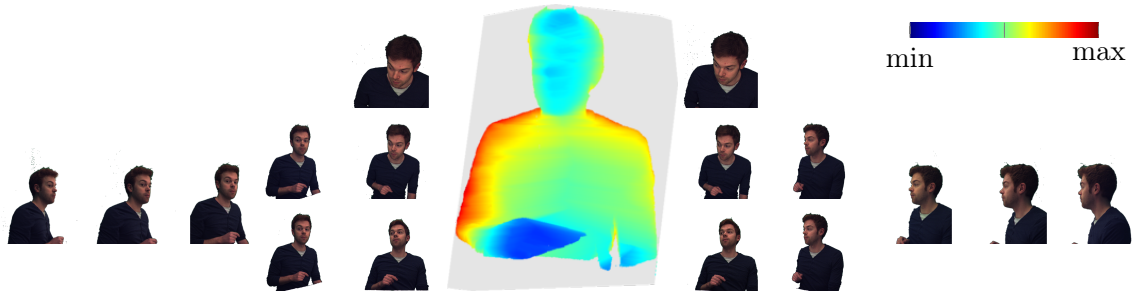


Figure 3.9: Silhouette images and color-coded depth map result for the `Niklas` dataset. The depth is computed for a reference view that is different from the 16 input views. The preselected image area is shaded gray.

3.1.5 Conclusion

In this section a substantially extended version of existing IBVH approaches was presented. The parallel algorithmic design allows for the processing of 16 HD input images on a single state-of-the-art graphics card in real-time. In addition, it was shown that the proposed pixel preselection can be considered as a beneficial approach to reduce the computational load. During experimental evaluation an average speedup of up to 45 percent was observed. Regarding a reasonable trade off between runtime and precision, instead of the original HD resolution, QHD or even QQHD can be used for the size of the reference image in order to save hardware resources while still maintaining a comparable depth precision. The experiments on the amount of cache bins showed that the optimal numbers of cache bins for all of these three resolutions are almost in the same range. Consequently, a high utilization of the graphics hardware is achieved without adapting the number of cache bins to certain resolutions. From a quality point of view, the proposed image border extension prevents an object cutoff in case a person is not completely visible in one or more silhouette images. Beside depth map computation, an orthographic camera configuration for the reference view allows for the generation of voxel equivalent representations with almost no additional computational costs.

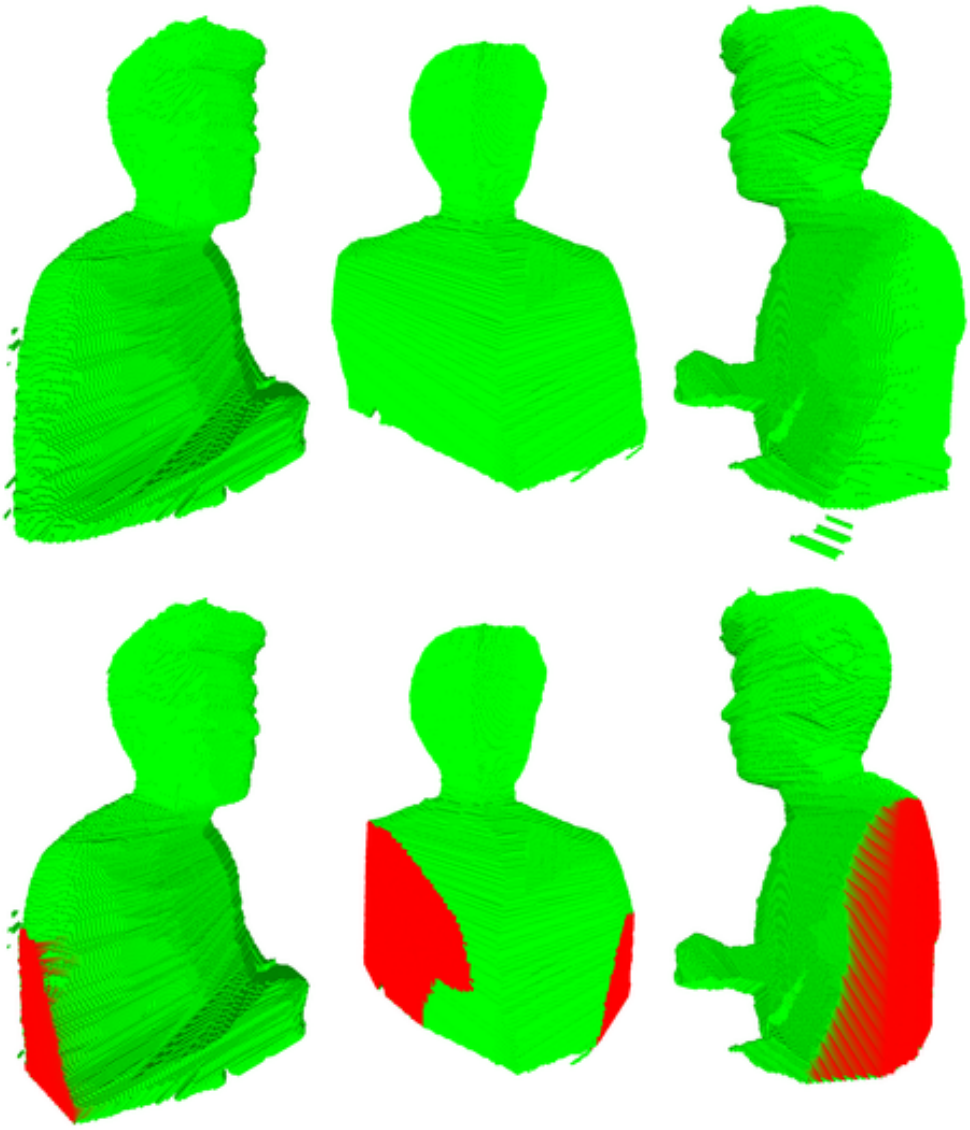


Figure 3.10: Voxel equivalent visualization of the result from figure 3.9. In the top row results with border extension are shown. The bottom row shows the very same results, but the object is cutoff at the image border. The cutoff areas are colored red.

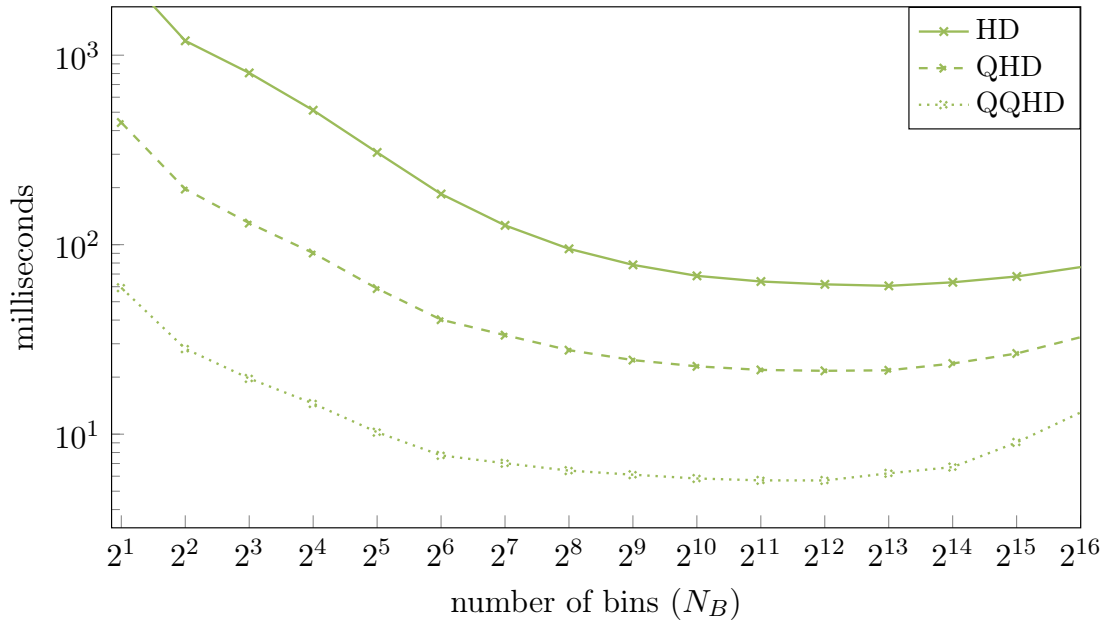


Figure 3.11: Impact of number of cache bins N_B on IBVH runtime for HD, QHD and QQHD resolutions. For each cache bin number an average timing result is computed on the `Niklas` dataset. Pixel preselection was activated for this experiment.

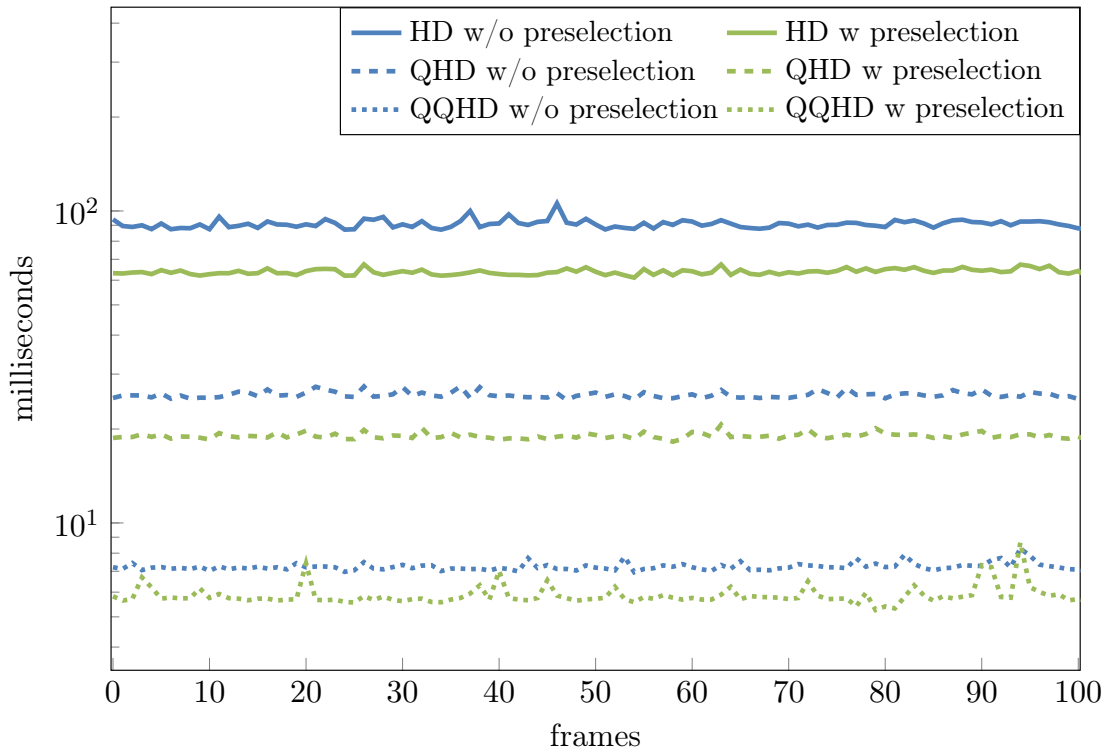


Figure 3.12: Runtime of VH computation for the `Niklas` dataset with different resolutions. For each resolution, there are timings with and without pixel preselection.

3.2 The Patch-Sweep Algorithm

The Patch-Sweep algorithm targets at a robust and accurate real-time 3D analysis for narrow baseline stereo camera systems and multi-view camera arrays, e.g. [BCP11]. The basic idea is to evaluate spatial 3D positions in terms of quantized rectangular patches. These patches are differently oriented in order to estimate the 3D object surface geometry as illustrated in figure 3.13. As the patches are estimates to the real surface, they are referred to as hypotheses in order to reflect their uncertain character. The approach is inspired by patch-based algorithms that are able to achieve very accurate results regarding the reconstruction of 3D objects in the domain of multi-view stereo [Sei+06b, FP07]. Here, the spatial patches are used as a linear approximation of the real object surface. In contrast to fixed matching windows that are used by many stereo algorithms like [RZK11], the application of spatial patches enables for a perspectively almost correct stereo matching. However, due to explicit occlusion handling and expensive seed and grow strategies, the computation time of patch-based approaches such as [FP10] can be situated in the range of minutes or even hours for a single frame [Sei+06a]. Regarding real-time performance, the Patch-Sweep algorithm conducts the evaluation of spatial patches individually for each pixel position and omits an explicit occlusion detection. This leads to an inherently parallel algorithmic design that is well suited for efficient implementation on parallel architectures like graphics hardware.

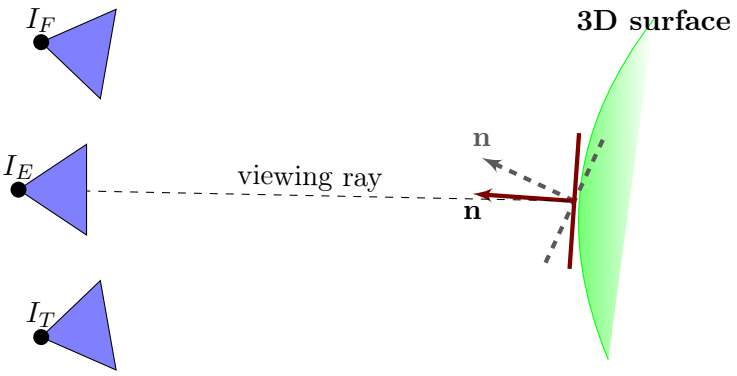


Figure 3.13: Algorithmic approach of the Patch-Sweep algorithm. Differently oriented patch prototypes are evaluated at various positions for the reference image I_E and the two input views I_F and I_T.

The section is structured as follows: A general perspective on the functionality of the Patch-Sweep algorithm together with the required notation that serve as reference for the following sections is provided in section 3.2.1. Subsequently, a naive variant of the Patch-Sweep algorithm that uses a straightforward exhaustive sweep strategy with a fixed depth range is presented in section 3.2.2. This very basic approach has a close relationship to early work on space-sweep algorithms [Col96]. While the exhaustive sweep is computationally expensive and becomes impractical for larger depth ranges, it serves well as a benchmark for the evaluation of the more elaborated Iterative Sweeping procedure as it is introduced in section 3.2.3. Here, the target is to reduce the computational load compared to the exhaustive sweep. For this purpose, an efficient Iterative Sweeping strategy is introduced. Compared to the exhaustive sweep, there is no fixed depth range. Instead, the Iterative

Sweep includes temporal predecessors from a spatial neighborhood that was computed in the previous iteration into the evaluation of spatial patches. Finally, experiments with synthetic and real-world data are conducted in section 3.2.4. The purpose of the experiments is threefold. First, the Iterative Sweep is benchmarked with respect to the naive exhaustive variant based on ground truth 3D data. Second, the results for stereo and trifocal camera configurations are compared. And third, the evaluation of a combination with IBVH results and a comparison with state-of-the-art multi-view and real-time stereo algorithms regarding qualitative aspects and algorithmic performance.

3.2.1 Algorithmic Approach

In the following, an overview to the proposed Patch-Sweep algorithm is provided. The required notation is introduced together with a formal definition of the applied patch representation. Constitutively, the evaluation of spatial patches is explained and the major algorithmic steps are presented. The Patch-Sweep algorithm operates on two or more input images denoted as stereo or multi-view camera input respectively. Optionally, corresponding foreground segmentation masks can be included. The reference view for which the 3D data is estimated is denoted as I_E while the additional input views are referred to as $\{I_{a_i}\}$, $a_i \in \{1, \ldots, m\} \setminus E$. For an image I, the corresponding image plane Ω_I is the set of valid image coordinates. The camera calibration information is represented in terms of a 3×4 projection matrix $\mathbf{P} = [\mathbf{KR} \,|{-}\mathbf{KRc}]$ that consists of a 3×3 rotation matrix $\mathbf{R}$, the camera center $\mathbf{c}$ and the intrinsic camera parameters

$$\mathbf{K} = \begin{pmatrix} f & s & u_x \\ 0 & \alpha f & u_y \\ 0 & 0 & 1 \end{pmatrix}, \tag{3.6}$$

where f is the focal length in pixel, α the pixel aspect ratio, s the skew parameter and $(u_x, u_y)^T$ denotes the principal point. In order to simplify arithmetic expressions and to reduce the computational load, the coordinate system of the reference camera I_E is aligned with the world coordinate system, i.e. $\mathbf{R}_E = \mathbf{I}$ and $\mathbf{c}_E = (0, 0, 0)^T$. In consequence, the projection matrix of camera I_E reads as

$$\mathbf{P}_E = (\mathbf{K}_E | \, \mathbf{0}) \,. \tag{3.7}$$

3.2.1.1 Patch Evaluation and 3D Surface Hypotheses

From a general perspective, the 3D analysis is carried out by the evaluation of differently oriented and positioned rectangular spatial patches that are referred to as hypotheses. Based on a similarity measure $\mathcal{S}$, the evaluation of a hypothesis is conducted with respect to the image pairs $I_E \times \{I_{a_i}\}$. For each pairwise evaluation, the sampled 3D values S^{Π} of a patch Π are determined by a block of $\mathbf{b}_x \times \mathbf{b}_y$ pixel coordinates that surround the projected patch center $\mathbf{x}_{pc} = (x_{pc}, y_{pc})^T \in \Omega_E$ as

$$S_E^{\Pi} = \left\{ \begin{pmatrix} x_{pc} - (\frac{\mathbf{b}_x}{2} - 0.5) + i \\ y_{pc} - (\frac{\mathbf{b}_y}{2} - 0.5) + j \end{pmatrix} \right\}_{(i,j) \in \{0,\ldots,\mathbf{b}_x - 1\} \times \{0,\ldots,\mathbf{b}_y - 1\}} . \tag{3.8}$$

These pixel coordinates are defined as the projection of the sampled 3D patch onto the image plane Ω_E. In consequence, the sampled 3D patch coordinates S^{Π} are given by the intersection of the viewing rays of S_E^{Π} with the 3D plane that contains the spatial patch. The depth value Z_{pc} of the patch center with respect to image I_E and the normal $\mathbf{n}$ of the patch are used to parameterize the perspectively correct transfer of S_E^{Π} onto a second image plane Ω_I. As the reference view I_E is aligned with the 3D world coordinate system, the depth based re-projection from $\mathbf{x}_{pc}$ to the 3D patch center $\mathbf{X}_{pc}$ can be expressed as

$$\mathbf{X}_{pc} = Z_{pc} \cdot \mathbf{K}_E^{-1} \begin{pmatrix} \mathbf{x}_{pc} \\ 1 \end{pmatrix}. \tag{3.9}$$

The homogeneous representation of the spatial plane that contains Π is given by $\left(\mathbf{n}^T, d\right)$, where $d = \langle \mathbf{X}_{pc}, \mathbf{n} \rangle$. In combination, the desired coordinate transfer for S_E^{Π} onto Ω_I is conducted with respect to the homography

$$\mathbf{H}(\mathbf{n}, Z_{pc}) = \mathbf{K}_I \left(\mathbf{R}_I - \mathbf{t}_I \mathbf{n}^T / d\right) \mathbf{K}_E^{-1} \tag{3.10}$$

that is induced by $\left(\mathbf{n}^T, d\right)$. In order to provide a coordinate transfer function for inhomogeneous image coordinates $\mathbf{x} = (x, y)^T$, the mapping

$$\mathcal{M}_{\mathbf{H}}(\mathbf{x}) := \begin{pmatrix} \frac{xh_{00}+yh_{01}+h_{02}}{xh_{20}+yh_{21}+h_{22}} \\ \frac{xh_{10}+yh_{11}+h_{12}}{xh_{20}+yh_{21}+h_{22}} \end{pmatrix}, \text{ with } \mathbf{H} = \begin{pmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{pmatrix} \tag{3.11}$$

is defined. The evaluation of a spatial patch is conducted via the comparison of the texture values for the patch coordinates S_E^{Π} in image I_E and the texture values for the transformed coordinates in image I as

$$\mathcal{S}\left(I_E\left(S_E^{\Pi}\right), I\left(\mathcal{M}_{\mathbf{H}(\mathbf{n}, Z_{pc})}\left(S_E^{\Pi}\right)\right)\right). \tag{3.12}$$

In this way, a perspectively correct matching is enabled and the probability of correctly reflecting the surface geometry is increased. As a secondary effect, not only the depth for each image coordinate is computed but also the surface normals of the object. For an illustration of spatial patch evaluation please refer to figure 3.13.

3.2.1.2 Major Algorithmic Steps

In order to provide a compact initial overview, the subsequent listing together with the diagram in figure 3.14 cover the generic algorithmic procedure that is the foundation for both, the exhaustive sweep and the Iterative Sweep. For all image coordinates $\mathbf{x} \in \Omega_E$ the following three basic steps are conducted:

1. A set of spatial patches that serves as hypotheses list is assembled.
2. Each patch is sampled according to section 3.2.1.1 and mapped to all images planes $\left\{\Omega_{I_{a_i}}\right\}$.
3. A similarity score according to equation (3.12) is computed for all image pairs $I_E \times$

$\{I_{a_i}\}$. The patch with the best score among all image pairs is stored as result. In case the optional foreground masks are provided, the worst score is assigned if the image coordinates of the patch center are part of the background for at least one view.

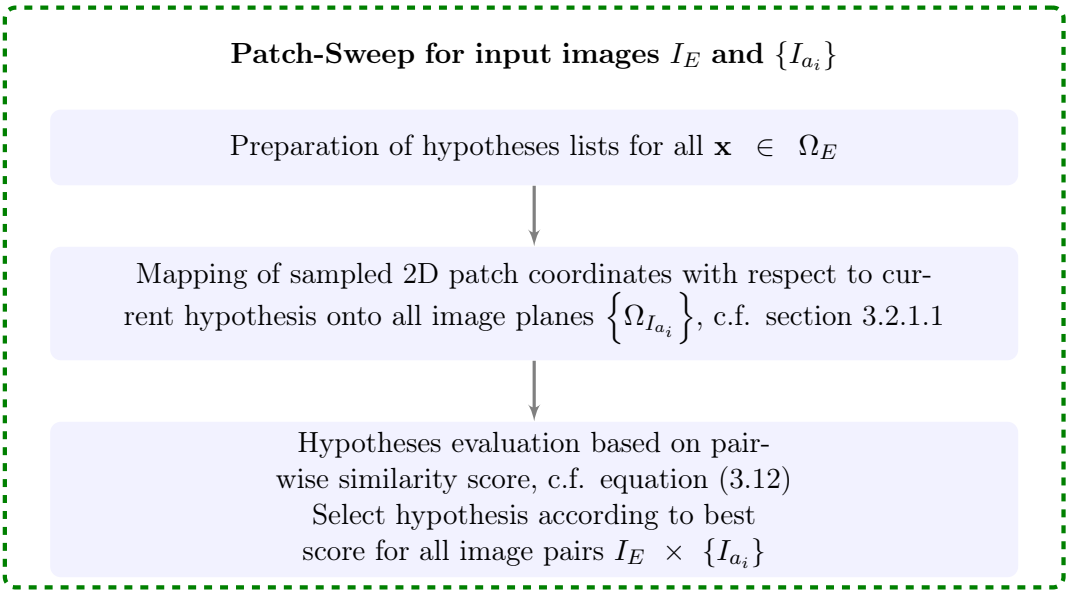


Figure 3.14: Structural overview of the algorithmic processing.

3.2.2 Exhaustive Sweep

In this section, the evaluation of depth hypotheses for a volume of interest is discussed. An exhaustive traversal with respect to a predefined quantization is proposed, and for each discrete 3D coordinate, a set of patches with different orientations is evaluated according to section 3.2.1.1. The required algorithmic components for volume quantization, patch orientation sampling and volume traversal are introduced as follows.

Volume Quantization The quantization is determined by the minimal and maximal depth values z_{min} and z_{max} and the number of discretization steps N_D. While the operational range can be set globally via fixed values for z_{min} and z_{max}, alternatively the depth map I_E^d from Visual Hull could be used to set a pixel-wise depth range according to

$$z_{min}(\mathbf{x}) = I_E^d(\mathbf{x}) \text{ and } z_{max}(\mathbf{x}) = I_E^d(\mathbf{x}) + z_r, \tag{3.13}$$

where z_r denotes a predefined depth range. For more details regarding this option, please refer to section 3.2.4.4. The set of discrete spatial coordinates are in line with the viewing rays of the reference view I_E. For each pixel coordinate $\mathbf{x}$ of image I_E its viewing ray $r := r(\mathbf{x})$ contains N_D spatial coordinates that are specified by the selected quantization scheme. The set of quantized 3D points $\mathbf{X}_i^r$ for each viewing ray is denoted as

$$Q^r = \left\{\mathbf{X}_0^r, \ldots, \mathbf{X}_{N_D-1}^r\right\}. \tag{3.14}$$

While there are many possibilities for volume discretization, in this work an elementary equidistant sampling with respect to $z-$direction is used. The alignment of the reference view according to equation (3.7) allows for a computationally efficient formulation for the computation of the individual 3D points $\mathbf{X}_i^r \in Q^r$ as

$$\mathbf{X}_i^r = \left(z_{min} + i \cdot \frac{z_{max} - z_{min}}{N_D - 1} \right) \cdot \mathbf{K}_E^{-1} \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix}. \tag{3.15}$$

Patch Orientation Sampling As an initialization step for the exhaustive sweep, a set of normals is computed that represent the possible patch orientations. For this purpose, it is required that the nature of the object's surface can be addressed by adjusting the initialization parameters. In case of a rough, jagged surface for example, extremal patch orientations are required in order to appropriately reflect the surface geometry. In order to provide a flexible solution for the computation of the different patch orientations, a unit hemisphere that resides in centered position on a frontoparallel patch Π is sampled according to three input parameters. The target is to uniformly distribute the sampling points on the parametrically selected part of the hemisphere. The lines connecting the center of Π and the sampled points on the surface of the hemisphere are the desired normals for representing different patch orientations. The three parameters are chosen as follows. First, the angle Λ, with $0 \leq \Lambda < \pi/2$ restricts the orientation to a maximal latitude. Second, $L_l > 0$ defines the total number of lines of latitude that are used for normal sampling. And third, the distance value d_{per} defines a constant longitudinal sampling distance for the perimeters of all lines of latitude. The sampling leads to $m := m(\Lambda, L_l, d_{per})$ different normal orientations. For an illustration of the sampling parameters please refer to figure 3.15. If $L_l = 1$, there is only one orientation normal $\mathbf{n}_0$. Otherwise, the set of orientation normals $\mathbf{N}^\Pi$ is computed as follows. Based on the input parameters Λ and L_l, the latitudinal step size $\Lambda_{step} = \Lambda/(L_l - 1)$ is identified. Regarding the i-th line of latitude, with $0 \leq i < L_l$, the perimeter per and the number of longitudinal steps H are given by $per(i) = 2\pi \cos(\pi/2 - i\Lambda_{step})$ and $H(i) := 1 + \lfloor per(i)/d_{per} \rfloor$ respectively. After computing the angular longitudinal step size $\theta_{step}(i) = 2\pi/H(i)$, the spherical coordinates of the sampling points reads as

$$s_{ij}^{sph} := (i\Lambda_{step}, j\theta_{step}(i)), \ 0 \leq i < L_l, \ 0 \leq j < H(i). \tag{3.16}$$

The bijective mapping of the spherical representation to Cartesian coordinates allows for a unique transformation to vector form $s_{ij}^{sph} \rightarrow \mathbf{c}_k$, with $0 \leq k < m$. The desired orientation quantization in terms of the orientation normals can be directly obtained by vector arithmetic

$$\mathbf{n}_k := \mathbf{c}_k - \mathbf{o}, \tag{3.17}$$

where $\mathbf{o}$ denotes the center of Π. Consequently, the desired set of patch normals reads as

$$\mathbf{N}^\Pi = \{\mathbf{n}_0, \ldots, \mathbf{n}_{m-1}\}. \tag{3.18}$$

Considering the algorithmic complexity, please note that for $L_l > 1$ the number of orientations can be analytically determined by

$$m(\Lambda, L_l, d_{per}) = \sum_{i=0}^{L_l - 1} 1 + \lfloor 2\pi cos(\pi/2 - i\Lambda/(L_l - 1))/d_{per} \rfloor \,. \tag{3.19}$$

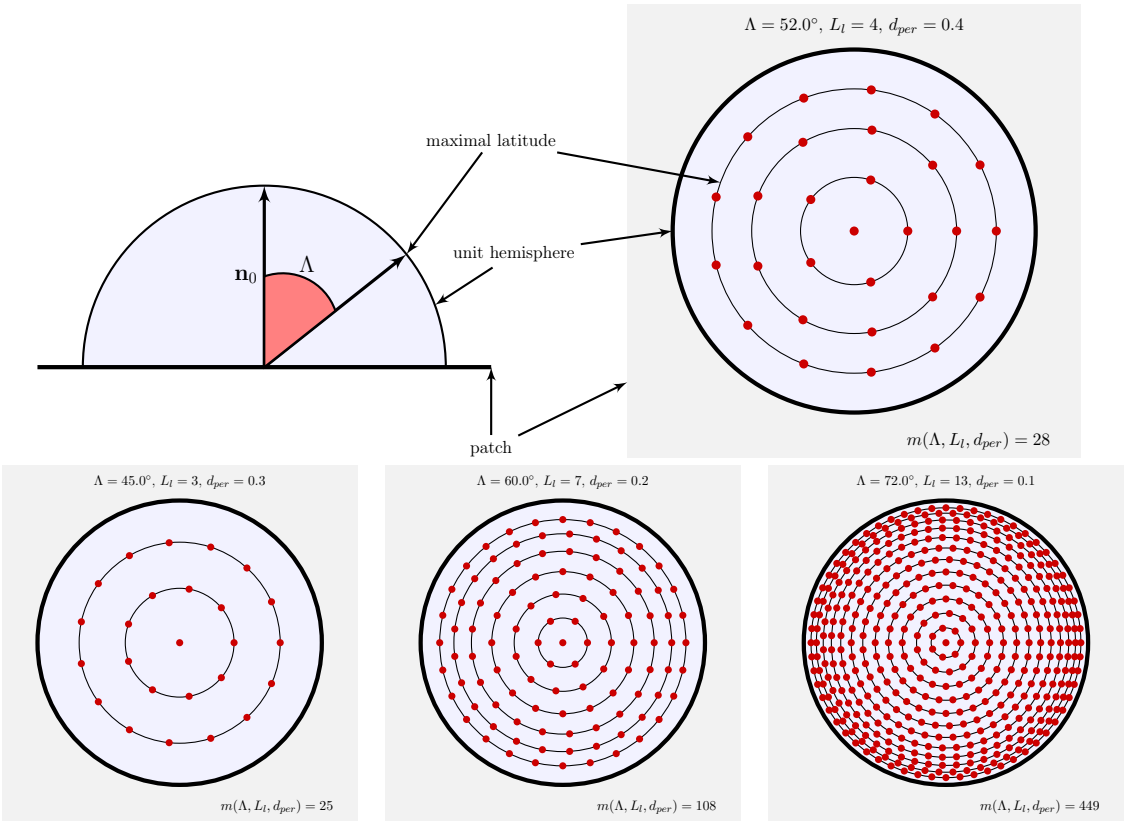


Figure 3.15: Patch orientation quantization via sampling of the unit hemisphere. **Top row:** Cross section of the hemisphere and the corresponding sampling points from above. **Bottom row:** Examples for hemisphere sampling with different values for Λ, L_l and d_{per}. The rightmost example shows the sampling that is used in the experiments section 3.2.4.1.

Volume Traversal and Matching The traversal of a single viewing ray r is an independent operation that can be done in parallel for all viewing rays. For each viewing ray a coarse to fine search is performed according to the number of refinement steps N_R and the refinement scaling parameter $s_R \in (0, 1)$. For each refinement step $j \in [0, \ldots, N_R - 1]$, a volume quantization as defined in equation (3.14) is computed according to equation (3.15). For the first step with $j = 0$ the quantization is computed via the initial depth range restriction $z^0_{min} = z_{min}$ and $z^0_{max} = z_{max}$. Then, for $j > 0$, the search range is restricted based on

the depth result from the last refinement step z^r_{j-1} as

$$z^j_{min} = z^r_{j-1} + \frac{d_j}{2} \text{ and } z^j_{max} = z^r_{j-1} - \frac{d_j}{2}, \tag{3.20}$$

where $d_j = (z_{max} - z_{min}) s^j_R$. For each of the resulting volume quantizations Q^r_j, the following procedure is performed in order to identify the best hypothesis.

For each pixel position $\mathbf{x} \in \Omega_E$ a hypotheses list is assembled. In compliance with section 3.2.1.1, the parameterization of a single hypothesis consists of a 3D point $\mathbf{X}^r_i \in Q^r_j$ and a surface normal $\mathbf{n}_k \in \mathbf{N}^\Pi$. In this context, please note that due to the aligned reference view I_E, the third component of $\mathbf{X}^r_i$ encodes the depth of the patch center with respect to I_E, c.f. equation (3.7). In consequence, the resulting hypotheses list contains all combinations of 3D points and normals as

$$L^{\mathcal{H}}_j(\mathbf{x}) = \left\{ Q^r_j \times \mathbf{N}^\Pi \right\}. \tag{3.21}$$

Each element of the hypotheses list is mapped to all image planes $\left\{ \Omega_{I_{a_i}} \right\}$ and evaluated with respect to a predefined similarity measure as discussed in section 3.2.1.1. In case the optional segmentation masks are available, a foreground check is performed as follows. If the patch center is contained in an image region that is not part of the foreground for at least one image, the worst matching score is assigned. If the image mask contains the patch center in all views, but the patch is partially not contained, the similarity measure is evaluated with respect to the patch values that reside inside the masked regions of the images, i.e. the computation is normalized with respect to the number of foreground pixel coordinates. The depth value of the patch center and the patch normal are extracted from the best hypothesis of $L^{\mathcal{H}}_j(\mathbf{x})$ and assigned to the image coordinates of the viewing ray. A structural illustration of the matching procedure is provided in figure 3.14.

3.2.3 Iterative Sweep

This section covers a framework for an iterative formulation of the sweeping procedure. While still following the general algorithmic approach as outlined in section 3.2.1, the target is to increase the algorithmic efficiency compared to the naive exhaustive sweep that was presented in section 3.2.2. The main idea is the integration of a spatial and temporal interdependency into the sweeping procedure while maintaining an efficient parallel execution on graphics hardware. The Iterative Sweep consists of a set of local rules for the iterative generation, update and propagation of hypotheses.

In the following, the Iterative Sweeping framework is presented and its convergence properties and performance are evaluated. In section 3.2.3.1, a structural overview to the algorithmic components is outlined. Based on the general structure diagram in figure 3.14, two diagrams are discussed that illustrate the generic building blocks of the Iterative Sweeping framework. Since the algorithmic framework allows for different sweeping variants, all components that are illustrated in figure 3.17 are discussed generically in the subsequent sections. Various configurations and compositions of these building blocks will be used in the course of different experiments in section 3.2.4. In section 3.2.3.2, details about the hy-

potheses generation and update are provided. Beside the formal derivation of deterministic hypotheses updates, a Monte Carlo sampling based hypotheses update is introduced. In addition to the patch-based representation of hypotheses that includes normal and depth information, simplified representations consisting only of depth or disparity are discussed. Depending on the concrete hypotheses representation, customized flavors of the algorithm can be derived that reflect application specific requirements regarding the computational demand of the algorithm and the completeness, smoothness, and accuracy of the results. In section 3.2.3.3, the selection of the spatial neighborhood from the previous iteration is described. It has a great impact on the overall algorithmic performance regarding convergence rate and quality. Thus, a change of neighborhood selection can be considered as the application of a different iteration scheme in terms of the propagation of the involved hypotheses. In section 3.2.3.4, a multi-scale approach is formulated that greatly reduces the computational load and improves the rate of convergence, especially in case of high resolution video input. Based on an empiric evaluation of real world datasets, an analysis of the rate of convergence for the Iterative Patch-Sweep is provided in section 3.2.3.5. Finally, preparative to the experiments section, a concise summary of the algorithmic configurations that are subject to evaluation is provided in section 3.2.3.6.

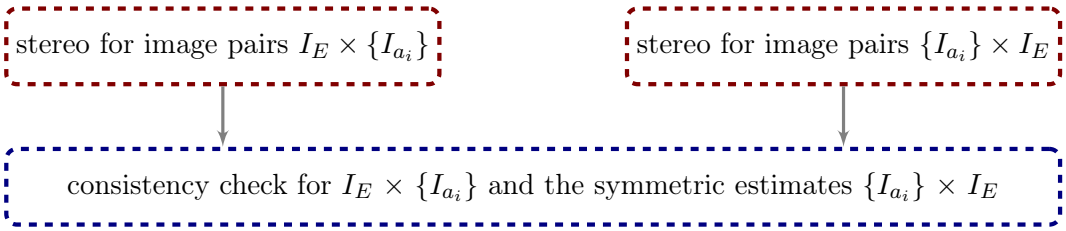


Figure 3.16: High level overview to 3D estimation and consistency check. A closeup of the estimation boxes (dashed red borders) is provided in figure 3.17.

3.2.3.1 Algorithmic Structure

The algorithmic components are structured as illustrated in figures 3.16 and 3.17. While figure 3.16 provides a high level overview to the processing blocks for 3D analysis and the subsequent consistency checks, figure 3.17 is devoted to a more detailed illustration of the Iterative Patch-Sweep procedure. The main idea for 3D estimation is the iterative evaluation and update of hypotheses in terms of spatial patches in order to approximate the depth and the normal of the real object surface for each pixel of image I_E. Regarding the processing of video sequences, the algorithm does not differentiate between a result for a previous frame and the intermediate state from a previous iteration. In this context, the iteration count t may refer to an iteration that was performed on a previous frame, and the value of t accounts for the sum of all previous iterations on all previous frames. Let $\mathcal{H}$ be the hypotheses map that contains the iteration results for each pixel position and $\mathcal{N}^L$ a generic definition for a hypotheses set from an L-neighborhood as

$$\mathcal{N}^L(\mathbf{x}, \mathcal{H}, t) := \{\mathcal{H}(\mathbf{x}_0, t), \ldots, \mathcal{H}(\mathbf{x}_{L-1}, t) \,|\, \mathbf{x} \neq \mathbf{x}_i\}\,. \tag{3.22}$$

The potential choices for the pixel positions $\mathbf{x}_i$ are introduced later on in section 3.2.3.3. For each iteration and each pixel coordinate $\mathbf{x} \in \Omega_E$, the *basic* list of hypotheses $L^{\mathcal{H}}$ consists of the hypothesis for the current position $\mathcal{H}(\mathbf{x}, t-1)$ from the previous iteration, several hypotheses from a spatial $L-$neighborhood $\mathcal{N}^L(\mathbf{x}, \mathcal{H}, t-1)$ of the current position from the previous iteration and an update $\mathcal{H}^u(\mathbf{x}, t-1)$ of $\mathcal{H}(\mathbf{x}, t-1)$. In combination, the *basic* hypotheses list reads as

$$L^{\mathcal{H}}(\mathbf{x}, t) = \left\{\mathcal{H}(\mathbf{x}, t-1), \mathcal{N}^L(\mathbf{x}, \mathcal{H}, t-1), \mathcal{H}^u(\mathbf{x}, t-1)\right\}. \tag{3.23}$$

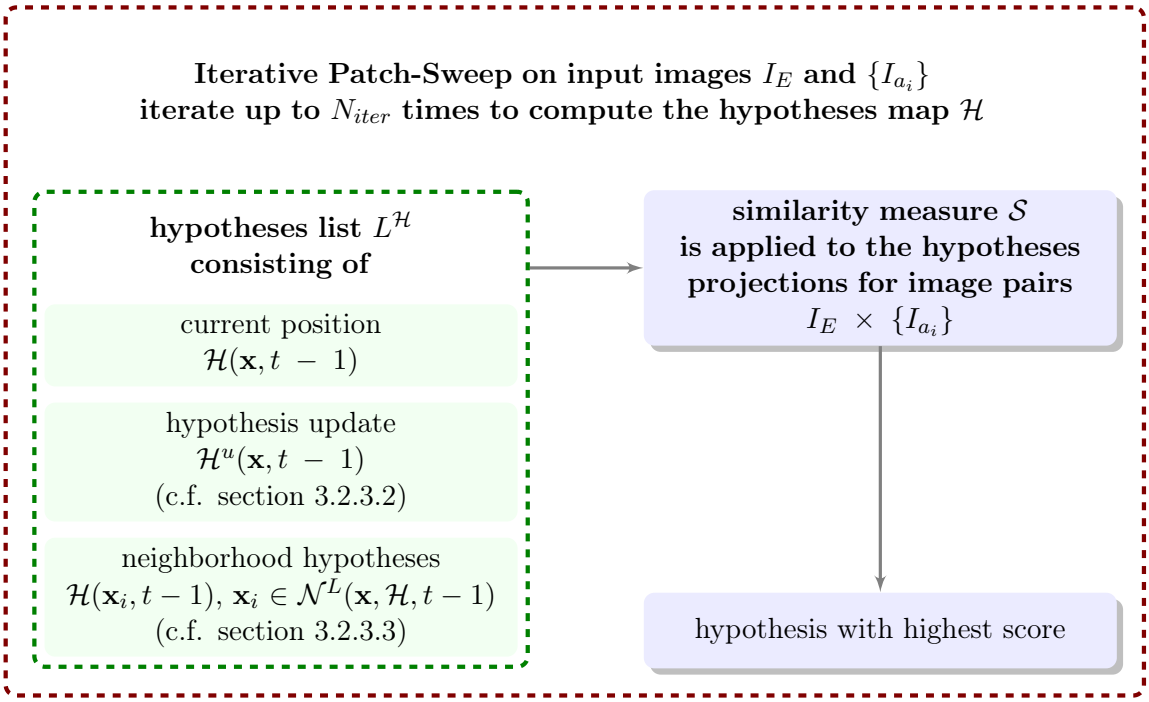


Figure 3.17: Algorithmic structure for the selection, the update, and the evaluation of hypotheses.

Hypotheses Representation In this thesis three different flavors of hypotheses representations are evaluated. Please note that this no complete list, but there is a huge amount alternatives. In general, surface elements of arbitrary shape and curvature would be possible. An overview to the selected hypotheses representations is listed in table 3.3. The hypotheses are represented with a parameter vector $\mathbf{p}$ and consist of *depth and normal orientation* $\mathbf{p} = (\mathbf{n}(a, b), Z)$, *depth only* $\mathbf{p} = (Z)$ and *disparity only* $\mathbf{p} = (\eta)$. In case of *depth only* hypothesis, two different variants are applied. While DE denotes a coordinate translation of an image patch, DEP respects the perspective distortion caused by patch projections that are conducted with the respective camera data.

Normal Parametrization The normal $\mathbf{n}(a, b)$ that is required for DEPN is parameterized as follows. The reference view I_E is assumed to be aligned with the world coordinate system as listed in equation (3.7), i.e. $\mathbf{P}_E = (\mathbf{K}_E | \mathbf{0})$ and $\mathbf{P}_I = \mathbf{K}_I (\mathbf{R}_I | \mathbf{t}_I)$ for an arbitrary

short	disparity	depth	normal	perspective	representation
DI	✓	–	–	–	$\mathbf{p} = (\eta)$
DE(P)	–	✓	–	(✓)	$\mathbf{p} = (Z)$
DEPN	–	✓	✓	✓	$\mathbf{p} = (\mathbf{n}(a,b), Z)$

Table 3.3: Overview to different hypotheses representations.

second view I. The normal $\mathbf{n}_{fp}$ of a fronto parallel patch with respect to $\mathbf{P}_E$ is oriented towards the camera, i.e. $\mathbf{n}_{fp} = (0,0,-1)^T$. Likewise, a valid normal of an estimated patch needs to exhibit a negative third component. A normal parameterization used for optimization (c.f. section 3.2.3.2) has to reflect the valid orientation directions. In this context, a mapping $\mathbf{n}(a,b) : \mathbb{R}^2 \rightarrow \mathbb{S}^2_l$ onto the lower hemisphere is used as

$$\mathbf{n}(a,b) = (a \cdot n_z, b \cdot n_z, n_z)^T, \text{ where } n_z = -\frac{1}{\sqrt{a^2+b^2+1}}. \tag{3.24}$$

Additional Hypotheses As equation (3.23) provides only a *basic* list of hypotheses, a supplementary *smoothness hypothesis* (SH) for DEPN is introduced as an example for the flexibility of the proposed approach. This hypothesis consists of the mean depth and the mean normal value from the current position and the $\mathcal{N}^4$ neighborhood of the previous iteration (c.f. section 3.2.3.3). The impact of this additional hypothesis is evaluated in section 3.2.4.1.

Parallelization and Functionality In order to allow for an efficient parallelization, the selection of the spatial neighborhood and the hypotheses update is strictly performed on pixel level without any reference to neighbors of the current iteration step. The concrete choice of the hypotheses update rule and the spatial neighborhood selection constitutes the functional efficiency of the iteration procedure. Although the Iterative Sweep does not comprise a global optimization criterion as with techniques like belief propagation, graph cuts, or variational approaches, the application of the local operations on pixel level leads to a comparable information propagation during the iteration. Once $L^{\mathcal{H}}$ is assembled, according to section 3.2.1.1 all hypotheses are evaluated for the image pairs $I_E \times \{I_{a_i}\}$ and the hypothesis with the best score is selected as new result.

Consistency and Completeness A consistency check for outlier and occlusion detection is applied as illustrated in figure 3.16. A consistency mask is generated that is based on the crosscheck of results for the symmetric estimations, i.e. for the image pairs $\{I_{a_i}\} \times I_E$. Once the hypotheses are computed, the corresponding symmetric estimates are compared with respect to the distance of the patch centers d_c. A hypothesis is considered to be consistent if $d_c < T_c$ holds for a predefined threshold T_c. In case of more than two views, a hypothesis is marked as consistent, if at least one crosscheck is successful. The fraction of consistent hypotheses is referred to as *completeness*. Considering post-processing of 3D data in terms of filtering or the fusion of multiple perspectives for view synthesis, as it will be discussed in section 3.2.4.5, the computed consistency mask constitutes a valuable cue for the reliability of hypotheses.

3.2.3.2 Hypotheses Update

In this section, the theoretical background for the update of hypotheses is covered in order to enable for an understanding of the functional efficiency of the algorithm. As illustrated in figure 3.17, the *hypotheses update* is the only operation that can lead to new hypotheses, while the other operations deal with the propagation of temporal predecessors from a spatial neighborhood that was computed in the previous iteration. During the Iterative Sweep, the hypotheses updates are based on a Monte Carlo sampling of possible update values. In order to benchmark the efficiency of the Monte Carlo sampling and to identify a suitable range of update values, a formulation for numeric optimization is derived that is used to compute deterministic hypotheses updates.

Numerical Optimization for Hypotheses Updates In the following, update procedures for the hypotheses representations that have been introduced in section 3.2.3.1 are covered. Preparatively, a generic optimization target for arbitrary hypotheses representations is formulated. Let $S_E^{\Pi} = (\mathbf{x}_0, \dots, \mathbf{x}_{N-1})^T$, $\mathbf{x}_i \in \Omega_E$ be the set of patch discretization coordinates according to equation (3.8) and $\mathcal{T}(\mathbf{x}, \mathbf{p}) : \Omega_E \times \mathbb{R}^M \to \Omega_I$ an image coordinate transformation between the image plane of I_E and $I \in \{I_{a_i}\}$. Each coordinate transformation $\mathcal{T}$ is induced by a certain hypothesis representation, where $\mathbf{p} = (p_0, \dots, p_{M-1})$ denotes the parameter set for the representation of a single hypothesis. Regarding equation (3.12), a valid hypothesis would lead to an optimal similarity score. For the numeric evaluation, the negated normalized cross correlation

$$\mathrm{NNCC}(\mathbf{X}, \mathbf{Y}) := -\mathrm{NCC}(\mathbf{X}, \mathbf{Y}) = -\frac{\left\langle \mathbf{X} - \overline{\mathbf{X}}, \mathbf{Y} - \overline{\mathbf{Y}} \right\rangle}{\|\mathbf{X} - \overline{\mathbf{X}}\|_2 \|\mathbf{Y} - \overline{\mathbf{Y}}\|_2} \tag{3.25}$$

is used as similarity score, where $\overline{\mathbf{X}}, \overline{\mathbf{Y}}$ are the means of the vectors $\mathbf{X}$ and $\mathbf{Y}$. In consequence, the criterion for hypotheses optimization is given by

$$\min_{\mathbf{p}} \left\{ \mathrm{NNCC}\left(I_E(S_E^{\Pi}), I(\mathcal{T}(S_E^{\Pi}, \mathbf{p})) \right) \right\}, \tag{3.26}$$

with short notations

$$\begin{aligned} I_E(S_E^{\Pi}) &:= (I_E(\mathbf{x}_0), \dots, I_E(\mathbf{x}_{N-1}))^T \text{ and} \\ I(\mathcal{T}(S_E^{\Pi}, \mathbf{p})) &:= (I(\mathcal{T}(\mathbf{x}_0, \mathbf{p})), \dots, I(\mathcal{T}(\mathbf{x}_{N-1}, \mathbf{p})))^T . \end{aligned} \tag{3.27}$$

The optimization is conducted via a gradient descent approach. Here, the parameter vector $\mathbf{p}$ is iteratively updated according to the direction of steepest gradient descent as

$$\mathbf{p}^{n+1} = \mathbf{p}^n - \alpha^n \boldsymbol{\lambda} \nabla f(\mathbf{p}^n), \tag{3.28}$$

where $f = \mathrm{NNCC}$ is the function that needs to be minimized and $\boldsymbol{\lambda}$ denotes a diagonal matrix containing the weights for the parameter vector update, c.f. equation (3.39). The

step size α^n is adjusted for each iteration by solving the subproblem

$$\min_{\alpha} f(\mathbf{p}^n - \alpha\boldsymbol{\lambda}\nabla f(\mathbf{p}^n)) \tag{3.29}$$

with an inexact line search according to the Armijo rule [Arm66] as follows. A fixed initial value is assigned to the current step size $\alpha^n \leftarrow \alpha_{\text{init}}$ and the inequality

$$f(\mathbf{p}^n - \alpha^n\boldsymbol{\lambda}\nabla f(\mathbf{p}^n)) \leq f(\mathbf{p}^n) - \alpha^n C\boldsymbol{\lambda}\nabla f(\mathbf{p}^n)^T\nabla f(\mathbf{p}^n) \tag{3.30}$$

is evaluated for a constant $C \in (0,1)$. If the inequality (3.30) holds, the step size is accepted. Otherwise it is iteratively updated according to a predefined $\beta \in (0,1)$ as $\alpha^n \leftarrow \beta\alpha_n$ until the inequality (3.30) holds for the first time. The substitution of f with NNCC from equation (3.26) leads to the required partial derivatives for the gradient computation as

$$\frac{\partial \text{NNCC}}{\partial p_j} = \sum_{i=0}^{N-1} \frac{\partial \text{NNCC}}{\partial I^i} \left(I_x \frac{\partial \mathcal{T}_x}{\partial p_j} + I_y \frac{\partial \mathcal{T}_y}{\partial p_j} \right)(\mathbf{x}_i), \tag{3.31}$$

where $I^i := I(\mathcal{T}(\mathbf{x}_i, \mathbf{p}))$, $\mathcal{T}(\mathbf{x}, \mathbf{p}) := (\mathcal{T}_x(x, \mathbf{p}), \mathcal{T}_y(y, \mathbf{p}))$ and $I_x = \frac{\partial I}{\partial \mathcal{T}_x}$ and $I_y = \frac{\partial I}{\partial \mathcal{T}_y}$ are the partial derivatives of image I in $x-$ and $y-$direction. In combination, the iterative parameter update is given by

$$\mathbf{p}^{n+1} = \mathbf{p}^n - \alpha^n\boldsymbol{\lambda} \left(\frac{\partial \text{NNCC}}{\partial p_0}, \ldots, \frac{\partial \text{NNCC}}{\partial p_{M-1}} \right)(\mathbf{p}^n). \tag{3.32}$$

In the following, based on the generic parameter update from equation (3.32), the iterative numeric parameter update procedures for the hypotheses representations as listed in section 3.2.3.1 are provide.

Disparity Update (DI) In case of rectified stereo images, numeric optimization can be conducted in the disparity space. From the perspective of the Iterative Patch-Sweep, it can be considered as a parameterization of fronto parallel patches in the disparity domain. Especially for application scenarios that are restricted to a single stereo camera configuration and disparity results, the algorithmic complexity is reduced due to the simple transfer function $\mathcal{T}(\mathbf{x}, \eta) = \mathbf{x} + (\eta, 0)^T$ and the single component parameter vector $\mathbf{p} = (\eta)$. Regarding equation (3.32), the parameter update for DI reads as

$$\eta^{n+1} = \eta^n - \alpha^n\lambda \sum_{\mathbf{x}_i \in S_E^{\Pi}} \frac{\partial \text{NNCC}}{\partial I^i} I_x(\mathbf{x}_i). \tag{3.33}$$

Depth Update (DE) If the estimation of depth for general camera configurations is required from a computational perspective, DE is the least expensive approach among the listing in table 3.3. The coordinate transfer is similar to DI. The spatial patches are considered to be fronto parallel with respect to camera $\mathbf{P}_E$. While the patch center $\mathbf{x}_{pc} \in \Omega_E$ is transformed with respect to the depth parameter $\mathbf{p} = (Z)$, the combined patch transfer does not reflect a perspectively correct projection. Instead, the normal parameter

of equation (3.10) is fixed to $\mathbf{n}_{fp}$. The resulting homography

$$\mathbf{H}(Z) := \mathbf{H}(\mathbf{n}_{fp}, Z) = \mathbf{K}_I \left(\mathbf{R}_I - \mathbf{t}_I \mathbf{n}_{fp}^T / Z\right) \mathbf{K}_E^{-1} \tag{3.34}$$

is used to define a mapping of inhomogeneous coordinates according to equation (3.11). Based on the transformation of the patch center, the patch is translated as $\mathcal{T}(\mathbf{x}, Z) = \mathcal{M}_{\mathbf{H}(Z)}(\mathbf{x}_{pc}) + (\mathbf{x} - \mathbf{x}_{pc})$. Regarding equation (3.32), the parameter update for DE reads as

$$Z^{n+1} = Z^n - \alpha^n \lambda \sum_{\mathbf{x}_i \in S_E^\Pi} \frac{\partial \mathrm{NNCC}}{\partial I^i} \left(I_x \frac{\partial \mathcal{T}_x}{\partial Z} + I_y \frac{\partial \mathcal{T}_y}{\partial Z} \right) (\mathbf{x}_i). \tag{3.35}$$

Depth Update with perspectively correct Matching (DEP) In contrast to DE, DEPN includes a perspectively correct patch transfer. While the parameter vector and $\mathbf{H}(Z)$ remain identical compared to DE, the transfer function is defined as a perspective mapping for all patch coordinates as $\mathcal{T}(\mathbf{x}, Z) = \mathcal{M}_{\mathbf{H}(Z)}(\mathbf{x})$. According to equation (3.32), the parameter update for DEP is performed as

$$Z^{n+1} = Z^n - \alpha^n \lambda \sum_{\mathbf{x}_i \in S_E^\Pi} \frac{\partial \mathrm{NNCC}}{\partial I^i} \left(I_x \frac{\partial \mathcal{T}_x}{\partial Z} + I_y \frac{\partial \mathcal{T}_y}{\partial Z} \right) (\mathbf{x}_i). \tag{3.36}$$

Depth and Normal Update with perspectively correct Matching (DEPN) The estimation of the patch normal and depth is carried out based on perspectively correct matching. DEPN is the most comprehensive surface patch representation that is listed in table 3.3. The estimation of normals enables for a better surface representation during the matching procedure and allows for a realistic shading during the rendering process. According to the normal parameterization of equation (3.24), the parameter vector reads as $\mathbf{p} = (a, b, Z)$. The possibility of varying normals is included to the transfer function by setting the normal parameter of equation (3.10) to $\mathbf{n} := \mathbf{n}(a, b)$. The resulting homography reads as

$$\mathbf{H}(a, b, Z) := \mathbf{H}(\mathbf{n}(a, b), Z) = \mathbf{K}_I \left(\mathbf{R}_I - \mathbf{t}_I \mathbf{n}^T / d\right) \mathbf{K}_E^{-1}. \tag{3.37}$$

Based on equation (3.11), the transfer function is given by $\mathcal{T}(a, b, Z, \mathbf{x}) := \mathcal{M}_{\mathbf{H}(a,b,Z)}(\mathbf{x})$ and the update of depth and normal parameters is carried out according to equation (3.32) as

$$\mathbf{p}^{n+1} = \mathbf{p}^n - \alpha^n \boldsymbol{\lambda} \sum_{\mathbf{x}_i \in S_E^\Pi} \frac{\partial \mathrm{NNCC}}{\partial I^i} (I_x, I_y) \begin{pmatrix} \frac{\partial \mathcal{T}_x}{\partial a} & \frac{\partial \mathcal{T}_x}{\partial b} & \frac{\partial \mathcal{T}_x}{\partial Z} \\ \frac{\partial \mathcal{T}_y}{\partial a} & \frac{\partial \mathcal{T}_y}{\partial b} & \frac{\partial \mathcal{T}_y}{\partial Z} \end{pmatrix} (\mathbf{x}_i). \tag{3.38}$$

Monte Carlo Sampling for Hypotheses Updates In practice, the downside of the gradient descent based numeric optimization is a significant computational demand for the evaluation of the similarity scores and their derivatives. In contrast, regarding real-time constraints, a Monte Carlo sampling of parameter update values can be efficiently realized with fast GPU based random number generators like the cuRAND library [NVI14] that are able to generate several gigasamples of random values per second. As the Iterative Patch-Sweep performs temporal propagation of spatially neighboring hypotheses, an individual

update value does not need to be exactly linked to the local texture values as with the iterative optimization. Instead, it is sufficient to have an amount of suitable update values for the different parameter sets $(\eta), (Z)$ and (a, b, Z) within a local neighborhood. These *seed values* are then distributed during each iteration. Therefor, a Monte Carlo approach is used to randomly sample possible update values. All random values are drawn from a (multivariate) standard normal distribution. Afterwards, similar to the gradient based update, a diagonal weighting matrix $\boldsymbol{\lambda}$ is applied to the update vector. In this context, the matrix multiplication leads to a change of the covariance of the normal distribution as $\Sigma_{\boldsymbol{\lambda}} = \boldsymbol{\lambda}^2$. Therefore, the weights of the matrix $\boldsymbol{\lambda}$ are denoted as σ_η, σ_Z and σ_{ab}. Consequently, the weighting matrices for the different hypotheses representations are given by

$$\boldsymbol{\lambda}_{\mathrm{DI}} = (\sigma_\eta),\ \boldsymbol{\lambda}_{\mathrm{DE(P)}} = (\sigma_Z) \text{ and } \boldsymbol{\lambda}_{\mathrm{DEPN}} = \begin{pmatrix} \sigma_{ab} & 0 & 0 \\ 0 & \sigma_{ab} & 0 \\ 0 & 0 & \sigma_Z \end{pmatrix}. \tag{3.39}$$

The weight entries of $\boldsymbol{\lambda}$ are empirically determined in order to maximize the update efficiency of the Monte Carlo approach in comparison to the gradient descent optimization. Preparative to the experimental section 3.2.4, suitable weighting values for the targeted video communication datasets will be provided in section 3.2.3.6.

3.2.3.3 Hypotheses Propagation

Referring to figure 3.17, for each iteration hypotheses of a spatial neighborhood that have been produced in the previous iteration are evaluated together with the updated hypothesis of the current location. Although there are only local decision rules involved, the resulting character of the proposed iterative hypotheses distribution is similar to algorithms that aim at the minimization of a global optimality criteria consisting of a data term and a smoothness term like with graph cuts, belief propagation, or variational techniques. Especially, the selection of the spatial neighborhood determines how hypotheses are propagated within the image plane. As will be shown by the evaluation in section 3.2.3.5, the selected neighborhood has a significant impact on the algorithmic behavior in terms of quality of the results and rate of convergence. According to the generic neighborhood description in equation (3.22), there is an infinite number of $L-$neighborhoods that can be chosen for the iteration procedure. Within this thesis, two different families of $L-$neighborhoods are investigated in section 3.2.3.5 in terms of quality of results and the rate of convergence. The first kind of $L-$neighborhoods is based on a deterministic preselection of involved coordinates according to a regular pattern. The second family is founded on a probabilistic selection of the neighborhood coordinates. An impression of the propagation behavior for the different neighborhoods is provided in figure 3.19. Here, intermediate results for the randomized neighborhood and for the deterministic neighborhood are illustrated.

Deterministic Spatial Neighborhood Many definitions of regular spatial neighborhoods can be found in literature. In order to provide a self-contained algorithmic description, the definition and notation are briefly outlined. As illustrated in figure 3.18, the neighbor-

hood formation happens in a regular manner. In the following, two different deterministic spatial neighborhoods are presented. While $\mathcal{N}^4$ will serve as a deterministic reference for the evaluation of the competing approach that is based on a randomized neighborhood selection, $\mathcal{N}^8$ is listed to provide at least one possible deterministic alternative. The definition of the $L-$neighborhoods for the deterministic case is given by a collection of a fixed set of neighboring coordinate values as follows:

$$\mathcal{N}^4(\mathbf{x}, \mathcal{H}, t) := \{\mathcal{H}((x+1, y), t), \mathcal{H}((x, y-1), t), \mathcal{H}((x-1, y), t), \mathcal{H}((x, y+1), t)\} \quad (3.40)$$

$$\mathcal{N}^8(\mathbf{x}, \mathcal{H}, t) := \{\mathcal{H}(\mathbf{x}_i, t) \,|\, \mathbf{x}_i \in \{\{x, x+1, x-1\} \times \{y, y+1, y-1\}\} \setminus \{x, y\}\}\,. \quad (3.41)$$

The potential propagation path length for each iteration is exactly one with respect to the metric that is induced by the selected neighborhood. $\mathcal{N}^4$ for instance induces the Manhattan distance, while $\mathcal{N}^8$ induces the Chebyshev metric. Hence, for a diagonal propagation, an iteration based on $\mathcal{N}^4$ requires two cycles, while $\mathcal{N}^8$-based iterations require only one cycle. In consequence, regarding the analysis of the algorithmic efficiency in terms of information propagation, the rate of convergence is directly linked to the choice of the spatial neighborhood that underlies the iteration procedure.

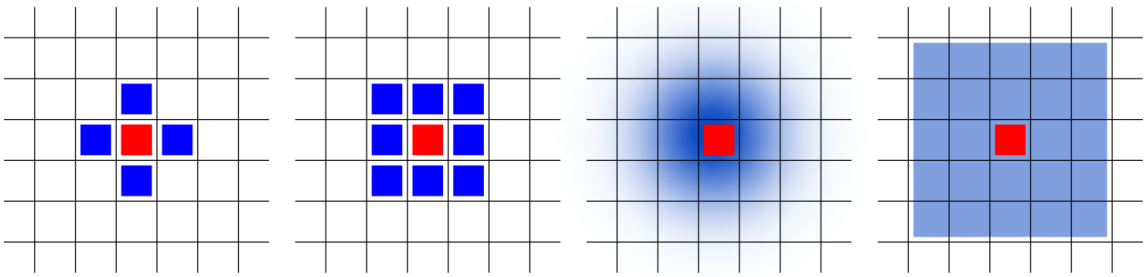


Figure 3.18: Illustration of different neighborhoods. From left to right: Deterministic 4− and 8−neighborhoods, randomized neighborhood with an underlying normal distribution, randomized neighborhood with an underlying uniform distribution. The red coordinate indicates the neighborhood center.

Randomized Spatial Neighborhood In contrast to deterministic spatial neighborhoods, the main idea of a randomized $L-$neighborhood $\mathcal{N}_r^L$ is the assembly of an individual neighborhood for each pixel coordinate and each iteration based on probability distributions. As illustrated in figure 3.18, the neighborhood definition consists of a likelihood for each coordinate position and a fixed number of samples r_s that are drawn randomly according to the underlying probability density function. In contrast to a deterministic neighborhood, the propagation path properties of a randomized neighborhood theoretically allow for a unlimited distance that can be covered within the course of a single iteration. In particular, the possible propagation distance is directly linked to the selected probability distribution. A uniform distribution, for instance, precisely restricts the potential distances, while a normal distribution allows for arbitrary propagation across the image plane. In this thesis, a bivariate normal distribution with zero mean and the covariance matrix $\Sigma_{\mathcal{N}_r^{rs}} = \begin{pmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_r^2 \end{pmatrix}$ is selected. There are two reasons for this choice. First, closer neighbors are considered to

share the same hypothesis more likely than distant neighbors. Second, as a normal distribution enables the propagation across large distances, there is a chance that good hypotheses values are spread within a single iteration. The adaptation of the covariance matrix $\Sigma_{\mathcal{N}_r^{r_s}}$ allows for a parameterization of the propagation behavior.

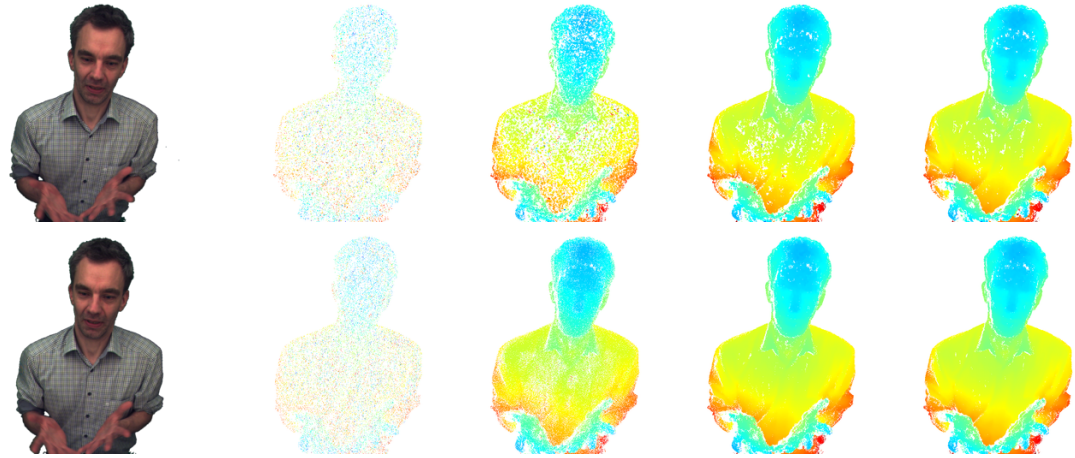


Figure 3.19: Two views of the `David` dataset as an example for the stereo input and intermediate results after iterations 1, 3, 6 and 9. The consistency threshold was set to $T_c = 10\,\text{mm}$. White areas indicate inconsistent depth values. **Left:** Stacked input images of the stereo view. **Top row:** Iterations with a deterministic 4−neighborhood. **Bottom row:** Iterations with a randomized neighborhood and $r_s = 4$.

3.2.3.4 Multi-scale Sweep

In order to increase the rate of convergence of the Iterative Patch-Sweep, a multi-scale strategy is formulated. With increasing image dimensions, the hypotheses propagation path lengths potentially expand. In consequence, with an increased path length an additional amount of iterations is required for hypotheses propagation. In order to speed up the propagation process, first solutions for low resolution input is computed and subsequently propagated as additional hypotheses for higher resolutions. Therefor, a number of sweep levels $L_s > 0$ are defined together with a level scale factor $0 < s_{lev} < 1$. For each level, scaled versions of the input images are computed with dimensions $w \cdot s_{lev}^l$, $h \cdot s_{lev}^l$, where w and h are the width and the height of the original images and $l \in [0, \ldots, L_s - 1]$ denotes the sweep level. Then, the Iterative Sweep is applied to each level in descending order, i.e. from the smallest image dimensions to the greatest. Please note that this formulation also implies a coarse to fine sweep strategy in 3D space. According to equation (3.8), the patch size is linked to pixel positions in image space. In consequence, the induced patch size in 3D is larger for higher sweep levels as a single pixel covers a greater amount of the scene in case of lower resolutions.

For each level, the resulting hypotheses map $\mathcal{H}^l$ is scaled by $\frac{1}{s_{lev}}$ in order to match the image dimensions of the next lower level $l - 1$. The required hypotheses interpolation is conducted in terms of nearest neighbor selection. Afterwards, $\mathcal{H}^l$ is propagated to level $l-1$ as an additional hypotheses input for the Iterative Sweep. In consequence, for $l < L_s - 1$

the multi-scale approach leads to an extension of the hypotheses list of equation (3.23) as

$$L_l^{\mathcal{H}}(\mathbf{x},t) = \left\{\mathcal{H}(\mathbf{x},t-1), \mathcal{N}^L(\mathbf{x},\mathcal{H},t-1), \mathcal{H}^u(\mathbf{x},t-1), \mathcal{H}^{l+1}(\mathbf{x},t)\right\}. \tag{3.42}$$

Compared to the other sweep levels, level 0 and level $L_s - 1$ are handled differently. On sweep level $L_s - 1$, no results from higher levels are available and the Iterative Sweep is conducted as without the multi-scale strategy. On level 0 there is no lower sweep level and the output constitutes the final result. Regarding complexity considerations, for each pixel of the input image a total of

$$k(s_{lev}, L_s) = (L+3)\sum_{l=0}^{L_s-1} s_{lev}^l - s_{lev}^{L_s-1} \tag{3.43}$$

hypotheses evaluations have to be conducted among all sweep levels. The impact of the multi-scale approach on the rate of convergence is evaluated within the course of the empiric convergence analysis in section 3.2.3.5.

3.2.3.5 Rate of Convergence

This section consists of three parts that are related to the rate of convergence. First, the algorithmic properties that are relevant for an efficient convergence are discussed. Second, the rate of convergence is analyzed empirically on real world data, and an empiric convergence criterion is proposed. And third, three different algorithmic variants are compared with respect to their impact on the rate of convergence. This evaluation includes multi-scale processing, the comparison of a deterministic neighborhood against its randomized counterpart, and the comparison of the numeric hypotheses update as described in section 3.2.3.2 with the Monte Carlo based hypotheses sampling.

Figure 3.20: Consecutive example frames for two input sequences that were recorded with 50 fps. **Top row:** Frames from the `David` dataset. This sequence exhibits very fast movements. Despite the high frame rate, there is a significant difference between consecutive frames. **Bottom row:** Frames from the `Oliver2` dataset. The person is moving moderately. Almost no difference is visible between consecutive frames.

Preliminaries on Algorithmic Characteristics As the Iterative Sweep exhibits no global optimization criterion, a converged state cannot be directly linked to a single numeric threshold that is related to some energy term. While the comparison of the 3D results between consecutive iterations or the computation of the mean squared error (MSE) or the peak-signal-to-noise ratio (PSNR) of the image that is mapped via the computed 3D data onto its stereo counterpart can serve as a general purpose indicator for a converged state, in the following, additional IPS specific criteria are identified by the analysis of algorithmic characteristics during the iteration procedure. First, the relevant parameters have to be identified. Regarding the algorithmic structure illustrated in figure 3.17, there are two operations that can affect convergence. These operations are the update and the propagation of hypotheses. The efficiency of hypotheses update operations is directly linked to the update acceptance rate. The more updates are accepted, the more *seeds* are generated and subsequently propagated. In this context, a hypothesis is considered as *seed*, if it is generated during a certain iteration or injected as a predecessor from the previous frame and remains at its position until convergence. The update of a hypothesis does not terminate its *seed* status, but the *seed* status is eliminated if the hypothesis is replaced by a propagated value. Please note that this procedure leads to a definition of *seeds* that can include wrong hypotheses. However, this does not contradict the objective since the conducted analysis is about convergence and not about correctness. The performance of the propagation operation can be expressed in terms of required iterations for propagating the *seed* hypotheses to all relevant positions. In combination, it is desirable to achieve a high *seed* generation rate and a small propagation path length according to the selected neighborhood. In order to formulate a convergence criterion on the basis of these requirements, the curves for the *mean propagation path length* and the *seed generation rate* are empirically evaluated. The focus is on the slopes of these curves, as their amplitude varies depending on the amount of scene dynamics. The following convergence analysis is conducted on two different datasets as shown in figure 3.20 in order to illustrate the impact of the persons' agility.

Empiric Analysis of the Rate of Convergence In figure 3.21, an empiric example for the fraction of accepted hypotheses updates, the *cumulated fraction of seed values*, and the *mean propagation path length* are plotted for five frames of the `David` and `Oliver2` datasets with 30 iterations for each frame. The intermediate results for the deterministic neighborhood illustrated in the top row of figure 3.19 correspond to the first frame of the dataset `David` in figure 3.21. In order to illustrate the asymptotic behavior of the measured values, the number of iterations is chosen much higher than required for a converged state and a stable result. The *hypotheses update fraction* refers to the ratio of accepted hypotheses updates and the overall amount of processed pixel positions. The *cumulated seed value fraction* is the fraction of hypotheses that keep their *seed* status from the current iteration until convergence. The *mean propagation path length* denotes the average iteration distance as described in section 3.2.3.3.

Although the plots reflect results for a certain dataset, the shapes of these graphs and especially their asymptotic behavior remain constant for all hypotheses representation that

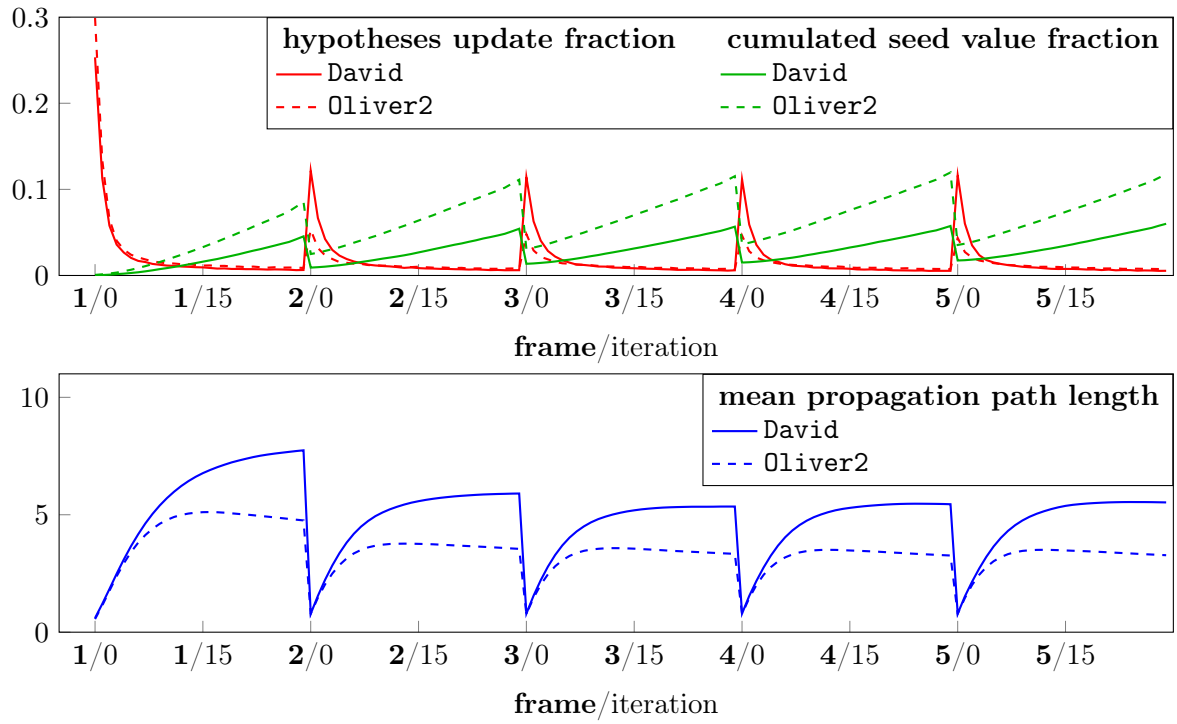


Figure 3.21: Statistics for the `David` and `Oliver2` datasets regarding hypotheses update acceptance, *seed* generation and propagation path length. The processing was conducted with the deterministic 4−neighborhood. For both datasets, it can be observed that with the progression of convergences, the hypotheses update fraction continuously decreases while the cumulated seed value fraction exhibits a linear growth. Simultaneously, the mean propagation path length increases until it reaches its upper bound and then decreases again due to little further changes from propagation operations.

were presented in table 3.3 and all evaluated datasets as listed in appendix B. Apparently, regarding the first frame, all three graphs are different compared to the consecutive frames. The reason for this deviation is the effort that is needed to iterate from an initial state towards the solution for the first frame. The iterations for subsequent frames are based on the previous results. Presuming a properly converged predecessor, the convergence rate is only affected by the hypotheses update and propagation efficiency, the choice of parameters and the dynamics of the scene content. By exemplarily comparing the estimation results for the dataset `David` in the top row of figure 3.19 for frame 1 at iteration 9 with the curves' progressions in figure 3.21, it can be seen that the results can be considered as converged as soon as the curve of the fraction of hypotheses updates and the curve of the *mean propagation path length* approach their lower and upper bound respectively. This observation is in compliance with the behavior that can be expected from the hypotheses selection rule, c.f. figure 3.17. As soon as all hypotheses updates are rejected, the existing hypotheses are considered as best choice. The saturation of the average propagation path length can be interpreted analogously. If the mean propagation path does not further expand, the existing hypotheses are superior to any competitor from a propagation attempt. An additional requirement for a fast convergence can be identified. Not only a small propagation path length is desirable, but also a steep ascent of the path length for each iteration until the upper

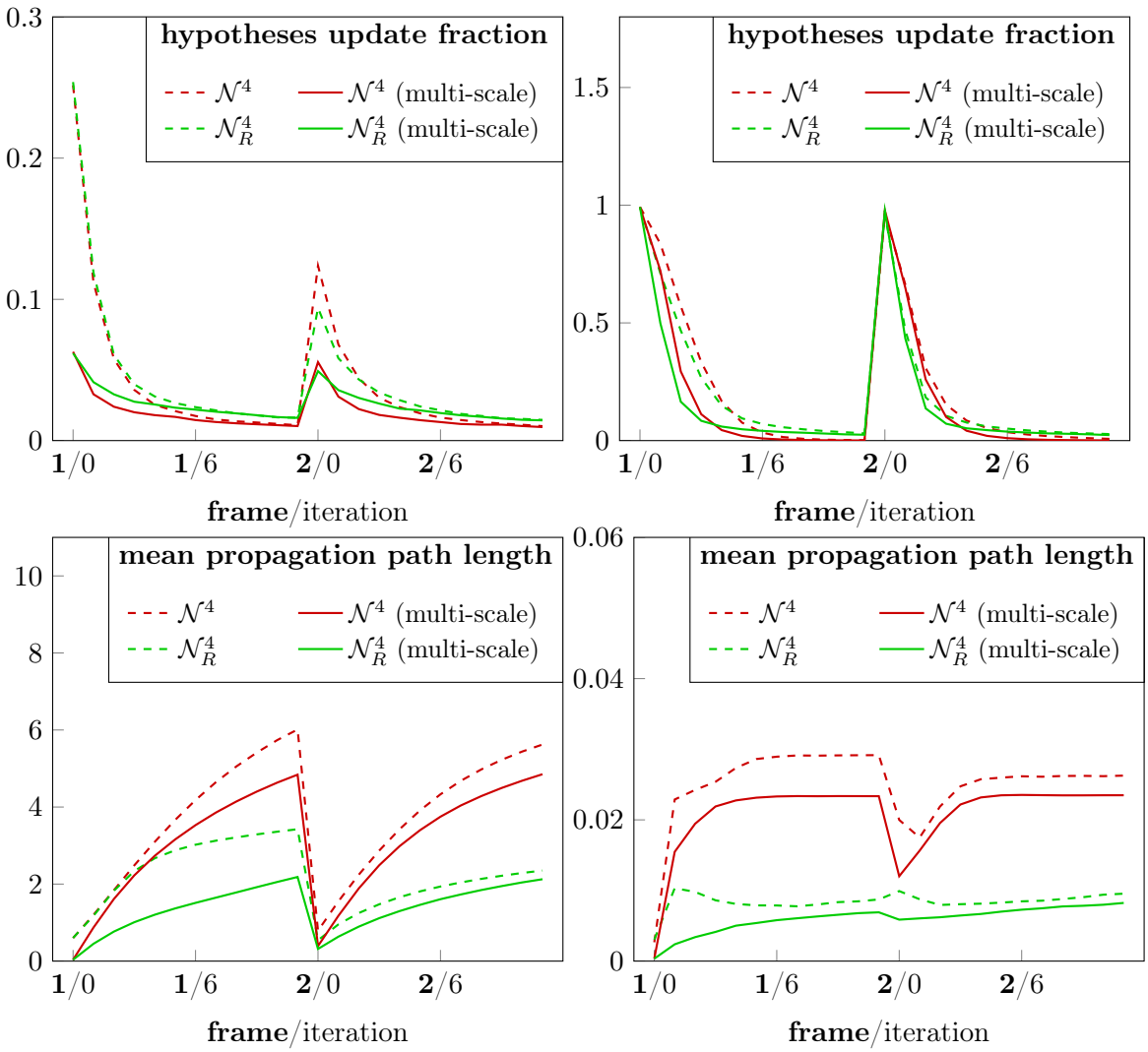


Figure 3.22: Comparison of convergence properties for different algorithmic components. **Left column:** Monte Carlo based hypotheses update. **Right column:** Numeric hypotheses update.

bound is reached. Finally, the *cumulated seed value fraction* plot can be interpreted as the empirically determined amount of *seed values*, that has been required to reach a converged state. According to figure 3.21, this value fluctuates approximately between 0.05 for the first frame and 0.11 for consecutive frames. Regarding a convergence criterion, the slope of the *mean propagation path length* can be considered as a robust cue for the achievement of convergence. The iteration could be terminated if this curve exhibits a predefined slope value or a maximal iteration count is exceeded.

Hypotheses Update and Propagation and Multi-scale Processing In the following, a comparison of the convergence properties between a deterministic 4− neighborhood and a randomized neighborhood $\mathcal{N}_R^4$ with four random samples is conducted and the impact of multi-scale processing with $L_s = 2$ and $s_{lev} = 0.75$ is investigated. According to equation (3.43), for each input pixel and each iteration there are $k(0.75, 2) = 10.375$ hypotheses evaluations with multi-scale and $k(s_{lev}, 0) = 6$ without multi-scale processing. Additionally,

the efficiency of the Monte Carlo based hypotheses update is evaluated with respect to the numeric hypotheses update as formulated in section 3.2.3.2. Figure 3.22 illustrates a closeup of the curves for the first two frames of the `David` dataset. In contrast to the analysis in figure 3.21, only 12 iterations have been carried out in order to better expose the slopes of the curves. Regarding the hypotheses update fraction, there is almost no difference between the randomized and the deterministic neighborhood. Due to the hypotheses updates from other sweep levels, the update fraction for the Monte Carlo based hypotheses updates is much lower for the first few iterations of the multi-scale sweep. Disregarding the hypotheses from multi-scale sweep, the initial acceptance rate for the Monte Carlo based updates is about 0.25, while the numeric hypotheses updates are almost completely accepted. As the computationally expensive numeric optimization is designed to find the best possible updates, the observation confirms the functional efficiency. The evaluation of the *mean propagation path length* reflects this situation comparably. The propagation path length is much shorter in case of numeric hypotheses updates. The high acceptance rate almost eliminates the requirement for hypotheses propagation.

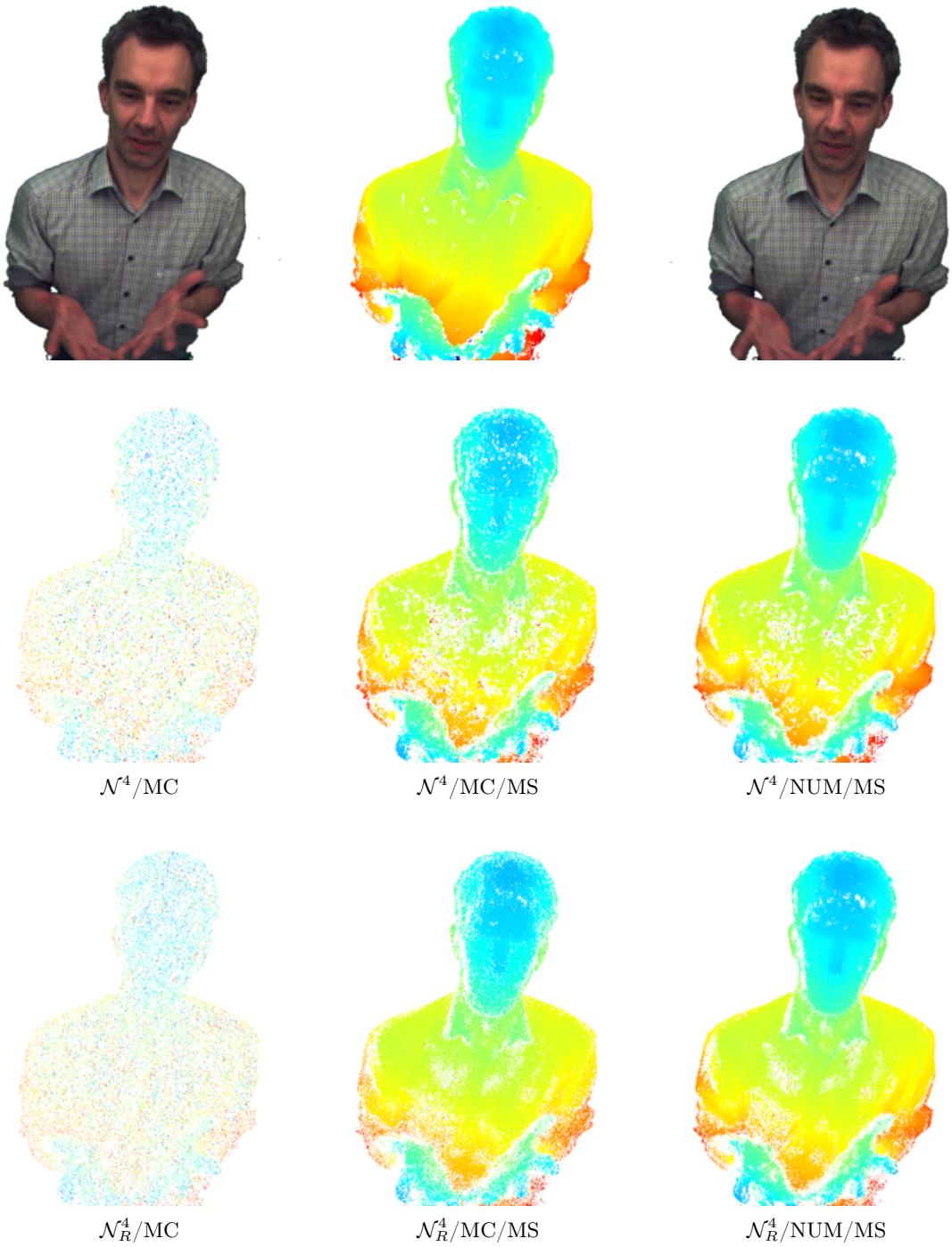


Figure 3.23: Input images, reference result and comparison of intermediate results for the first iteration of algorithmic configurations that are illustrated in figure 3.22. **First row:** Left input image, EPS reference result, right input image. **Second and third row:** Results for the deterministic neighborhood $\mathcal{N}^4$ and the randomized neighborhood $\mathcal{N}_R^4$, Monte Carlo based hypotheses updates (MC) and numeric hypotheses updates (NUM) and with multi-scale sweep (MS).

In case of Monte Carlo based updates, a significant difference between the randomized and the deterministic neighborhood can be observed for the *mean propagation path length*. In case of $\mathcal{N}^4$, the average path length is almost twice as large as with the randomized neighborhood. Additionally, as the effect of arbitrary path lengths for $\mathcal{N}_R$ cumulates across

the sweep levels, the effect of path length reduction for the multi-scale sweep is considerably greater for $\mathcal{N}_R^4$. This observation confirms the considerations regarding propagation path lengths in section 3.2.3.3. As the randomized neighborhood theoretically allows for a propagation across arbitrary distances, fewer iterations are required to propagate the hypotheses, and the resulting mean propagation path length is smaller compared to the deterministic variant. For all evaluated datasets, the mean propagation path length ratio between $\mathcal{N}_R^4$ and $\mathcal{N}^4$ is approximately $1 : 2$.

A qualitative comparison of the rate of convergence is provided in figure 3.23. The figure shows the left and right input images, the reference result that was computed with EPS and six intermediate results corresponding to the first iteration of the first frame of figure 3.22. Here, the first frame is selected to illustrate the worst case scenario. The initial hypotheses are random and the results are computed without any knowledge about previous frames. By comparing the intermediate results without and with multi-scale sweep, for the latter a significant improvement of the rate of convergence can be observed. The qualitative comparison simultaneously reveals two major benefits of the randomized propagation approach. First, especially in case of multi-scale sweep, the rate of convergence is greater than for the deterministic neighborhood. Second, compared to the deterministic neighborhood there is almost no speckle noise as the randomized structure of the neighborhood mitigates the local accumulation of mismatches. It is also notable that despite starting completely uninitialized already after the first iteration the $\mathcal{N}_R^4$/MC/MS result is very close to the EPS reference. In addition there is almost no difference between the $\mathcal{N}_R^4$/MC/MS and the computationally very expensive $\mathcal{N}_R^4$/NUM/MS results. The efficient hypotheses propagation compensates for the lower hypotheses acceptance rate of the Monte Carlo sampling.

		Hypothesis		
	Neighborhood	Representation	Update	$L^{\mathcal{H}}$
IPS-DI	$\mathcal{N}_R^4$	DI	Monte Carlo	*basic*
IPS-DEP	$\mathcal{N}_R^4$	DEP	Monte Carlo	*basic*
IPS-DEPN	$\mathcal{N}_R^4$	DEPN	Monte Carlo	*basic*
IPS-DEPN+SH	$\mathcal{N}_R^4$	DEPN	Monte Carlo	*basic* + SH

Table 3.4: Combinations of Iterative Sweep components that are subject for evaluation.

3.2.3.6 Combinations of Components and Choice of Parameters

In the previous sections, the properties of different interchangeable components for the Iterative Sweeping framework were discussed individually. While there are many possibilities to combine the presented components in order to form a concrete sweeping algorithm, the subsequent evaluations will only cover a manageable subset. In section 3.2.3.5, the benefits of multi-scale processing were highlighted. In addition, the efficiency of the Monte Carlo based hypotheses sampling and the randomized neighborhood could be confirmed. Constitutively, a compact summary of the combined components that will be evaluated in section 3.2.4 is provided in table 3.4. The *basic* hypothesis list refers to the definition in equation (3.23) and the algorithmic structure in figure 3.17. In order to illustrate the

impact of additional hypotheses, the supplementary *smoothness hypotheses* (SH) that was introduce in section 3.2.3.1 is additionally applied. In the experiments section, IPS-DEP, IPS-DEPN, and IPS-DEPN+SH are evaluated in section 3.2.4.1 and IPS-DI is evaluated in section 3.2.4.3. Additionally, IPS-DEP is used for the stereo versus trifocal comparison in section 3.2.4.2.

Based on the selected components, the optimal parameter set for the targeted video communication content is identified. Therefor, an extensive range of parameters is evaluated and the outcomes are compared with reference results from EPS. An overview to value ranges and the finally selected parameters is provided in table 3.5. As it was shown in sec-

parameter	**range**	**best**	**trade-off**	**description**
N_{iter}	1	1	1	maximal number of iterations
T_c	10	10	10	patch center distance threshold (mm)
$\mathbf{b}_x \times \mathbf{b}_y$	9×9	9×9	9×9	patch size in pixel
$\mathcal{S}$	NCC	NCC	NCC	similarity measure
r_s	$\{2, 4, 6, 8, 10\}$	6	4	number of random hypotheses
σ_r^2	$\{1, 2, 3, 4, 5, 6, 8, 10\}$	4	4	diagonal values of $\Sigma_{\mathcal{N}_r^{r_s}}$
σ_{ab}^2	$\{0.0625, 0.25, 0.5, 1, 2, 3\}$	0.125	0.125	normal update related values of $\Sigma_{\boldsymbol{\lambda}}$
σ_z^2	$\{1, 3, 5, 7, 9, 11\}$	5	9	depth update related value of $\Sigma_{\boldsymbol{\lambda}}$
L_s	$\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$	9	2	sweep levels
s_{lev}	$\{0.25, 0.5, 0.75\}$	0.75	0.75	sweep level scale factor

Table 3.5: Fixed parameters and value ranges for the evaluation of adequate Patch-Sweep parameter settings. All combinations of parameters were evaluated. Regarding the multi-scale parameters, only the combinations that lead to an image size greater than 32^2 pixel are used. The best parameters and the trade-off parameters are listed in separate columns. In this context trade-off reefers to the best configuration that can still maintain real-time processing.

tion 3.2.3.5, in case of multi-scale processing a good result can be computed within a single iteration. For this reason, the objective of the parameter evaluation was the identification of the parameter set that achieves the fastest convergence within one iteration under the constraint of maintaining a consistency threshold of $T_c = 10\,\text{mm}$. All results were computed on different frames of the `David` dataset, but without any information from previous frames. The `David` dataset was selected to enable the EPS for the production of reasonable benchmark results. As it is shown in table 3.6, the clothes that were worn in other sequences cause mismatches due to matching ambiguities that originate from their homogeneous structure. For evaluation, the IPS depth maps for the various configurations are compared to the EPS results with respect to their absolute difference, their completeness and their confidence in terms of the mapped similarity score $\frac{\text{NCC}+1}{2} \in [0, \ldots, 1]$. The absolute difference was computed for all pixel positions that are marked as consistent in the IPS and the EPS result. The EPS reference results exhibit an average completeness of 93.62 percent with respect to

the consistency threshold. The best result within the thousands of evaluated configurations reached an average completeness of 82.93 percent and a mean absolute difference with respect to the EPS reference results of 3.51 mm. The parameters that enabled for this result are listed in table 3.5. While these parameter values enable for the top results, there are other configuration that are very close, but require significantly less sweep levels L_s and random samples r_s. For this reason, a separate trade-off configuration that targets on the balancing between computational complexity and quality of results is listed in table 3.5. This configuration reaches an average completeness of 80.64 percent and a mean absolute difference of 3.79 mm.

Figure 3.24: Results for the `Marcus` dataset that illustrate the completeness values for homogeneous image regions. **From left to right:** Left input frame, EPS results, IPS results. The white areas indicate inconsistent depth values.

On basis of the identified parameter configurations in table 3.5, results for all the datasets with sixteen views that are listed in appendix B where computed, and the information from previous frames are regularly used to populate the hypotheses list as discussed in section 3.2.3.1. Exemplarily, the frame wise performance of both configurations on the `David` dataset is illustrated in figure 3.25 and average values for different datasets are shown in table 3.6. For all datasets, the trade-off results are very close to the results of the best configuration. Therefore, the trade-off configuration constitutes an adequate parameter selection. While eps exhibits the best confidence due to its full search strategy, it can be seen that the IPS outperforms the EPS results regarding the completeness percentage. This is caused by the homogeneous regions in the clothes of the conferees. As the exhaustive sweep evaluates much more hypotheses than the Iterative Sweep, there are significantly more matching ambiguities that can lead to mismatches. At the same time, IPS propagates hypotheses from previous frames within a spatial neighborhood and benefits from this guidance in terms of a higher matching consistency. An example frame from the `Marcus` dataset is shown in figure 3.24.

	completeness			confidence			absolute difference	
	EPS	best	trade-off	EPS	best	trade-off	best	trade-off
David	93.87	94.07	93.79	0.9637	0.9522	0.9518	2.21	2.29
Sylvain	53.81	67.75	66.82	0.9104	0.8937	0.8942	7.75	7.57
Marcus	56.23	79.09	78.16	0.8765	0.8608	0.8612	8.78	8.63
Paul	87.82	92.59	92.29	0.9079	0.8949	0.8948	2.35	2.39
Niklas	88.09	93.30	93.02	0.8745	0.8637	0.8636	2.73	2.74
Oliver2	88.18	92.44	92.16	0.9180	0.9083	0.9082	2.55	2.57

Table 3.6: Dataset wise average completeness values and absolute differences with respect to the EPS reference results. The respective configurations are listed in table 3.5.

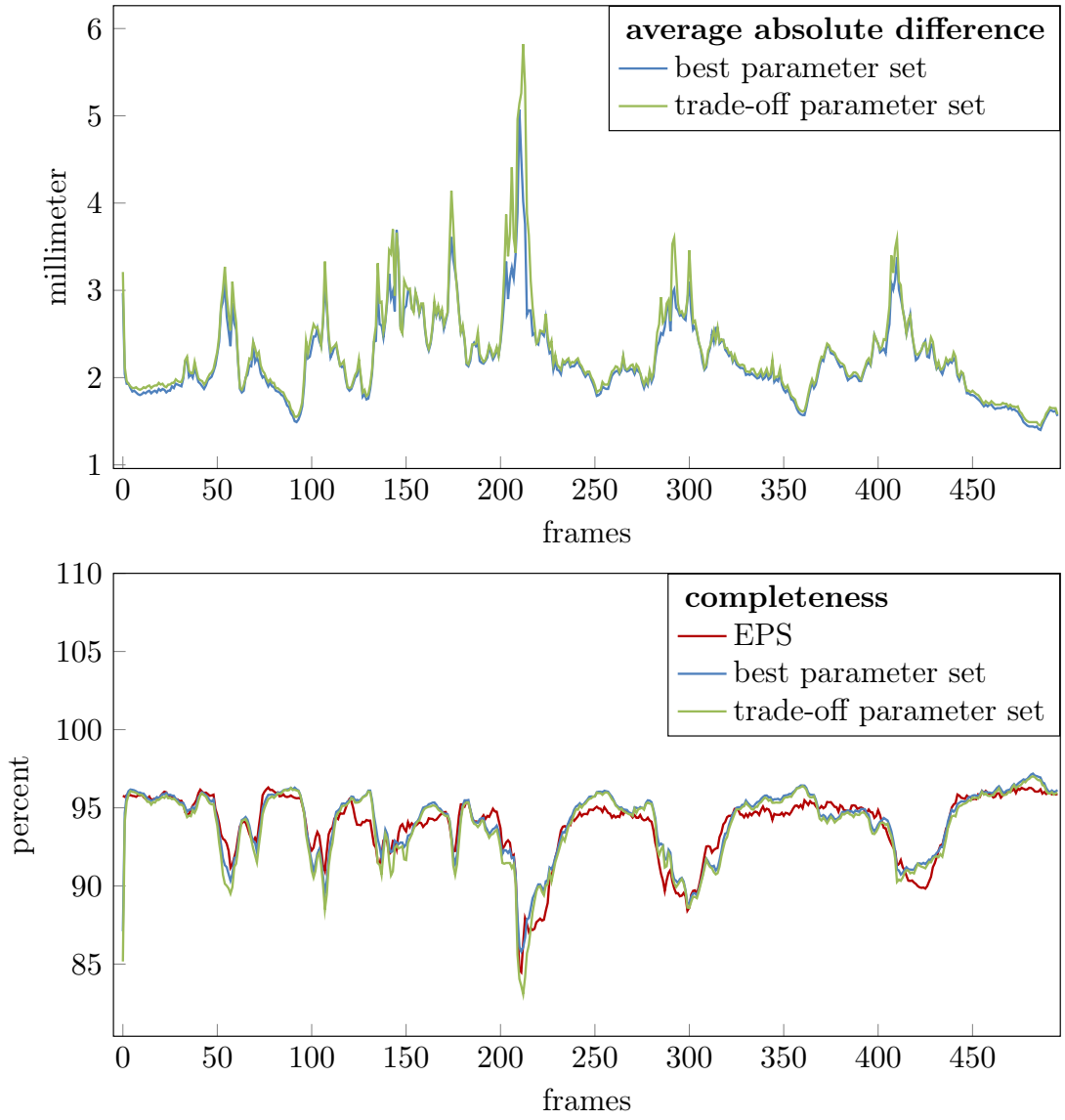


Figure 3.25: Completeness and average absolute difference for the David dataset. The evaluation was conducted with respect to the EPS reference results. The *best parameter set* and the *trade-off parameter set* are listed in table 3.5.

3.2.4 Experiments

In the following experiments, different aspects of the Patch-Sweep algorithm are evaluated on synthetic and real-world data. An overview to the real-world datasets can be found in Appendix B. All of these datasets were recorded with the demonstrator setup as it is described in Appendix A. The stereo baseline of the cameras was approximately 100 mm for all datasets.

The experimental evaluation starts with a comparison of different Patch-Sweep variants in section 3.2.4.1. Afterwards, the benefit of trifocal configurations compared to pure stereo setups is analyzed in section 3.2.4.2. As the envisaged video communication scenario implies real-time constraints and requires high quality results, a comparison to state-of-the-art real-time algorithms in terms of algorithmic efficiency and quality is conducted in section 3.2.4.3. In section 3.2.4.4 a combination of IBVH and Patch-Sweep is evaluated with a focus on the completeness of the results in case of less textured image regions. Finally, in section 3.2.4.5 the individual Patch-Sweep results from different views are combined and the results are qualitatively evaluated with respect to a popular state-of-the-art multi-view 3D reconstruction work-flow.

3.2.4.1 Iterative Flavors versus Exhaustive Sweep

In this section, the focus is on the evaluation of the three depth based flavors of the Iterative Sweep as listed in table 3.4 and the exhaustive sweep as described in section 3.2.2. The objective is to compare the quality that is achievable with different hypotheses representations among each other and against the brute force strategy that is used with the exhaustive sweep. In addition, the algorithmic efficiency of the iterative approach compared to brute force sweeping is evaluated. In order to enable for a numeric comparison, synthetic ground truth 3D data in combination with a real-world texture is used to render a left and a right input view. The synthetic camera pair for object rendering is in a rectified state and has a baseline of 40 mm and a focal length of 1600 pixel. The synthetic 3D test object has a distance from the cameras of about 900 mm and consists of a slanted plane that is intersecting a sphere with a radius of 100 mm as shown in figure 3.26. The choice of this object and the texture is motivated by two reasons. First, the shape of the object unifies three interesting surface geometries: a smooth linear surface, a curved surface that cannot be properly approximated with fronto parallel patches and an almost 90 degree kink at the transition between the sphere and the plane. Second, the selection of the real-world texture targets on the minimization of potential mismatches. In this way, the choice of the similarity measure has a minor impact, while the focus is on the comparison of the efficiency of the sweeping procedures.

The parameter settings for all experiments are chosen as shown in table 3.7. The parameters for EPS and the IPS variants have been empirically adjusted. Especially in order to perform a meaningful comparison with respect to the implicit support of IPS for arbitrary small depth and normal quantization, the quantization related parameters for EPS, N_D, N_R, s_R, Λ, L_l and d_{per} have been carefully tuned until there was no further quality improvement. According to equation (3.19), the resulting parameter set leads to a number

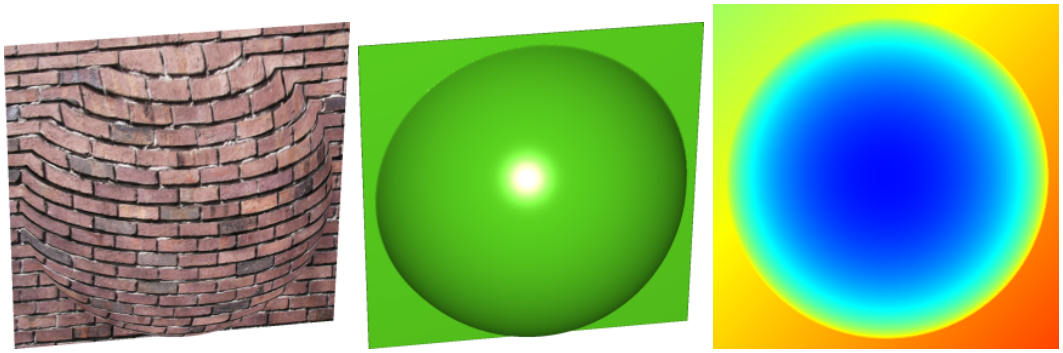


Figure 3.26: Synthetic test object with real-world texture. **Left:** Textured 3D object. **Middle:** Rendered 3D object surface. **Right:** Ground truth depth map for the left input image.

relevance	parameter	value	description
z_{min}	850	EPS	initial begin of depth range (mm)
z_{max}	980	EPS	initial end of depth range (mm)
N_D	31	EPS	depth discretization steps
N_R	5	EPS	coarse to fine steps
s_R	0.2	EPS	coarse to fine interval scaling
Λ	1.25	EPS	maximal latitude value
L_l	13	EPS	lines of latitude
d_{per}	0.1	EPS	metric perimeter quantization
N_{iter}	∞	IPS-ALL	maximal number of iterations
T_c	∞	IPS-ALL	patch center distance threshold (mm)
r_s	4	IPS-ALL	number of random hypotheses
σ_r^2	10	IPS-ALL	diagonal values of $\Sigma_{\mathcal{N}_r^4}$
σ_{ab}^2	1	IPS-DEPN(+SH)	normal update related values of $\Sigma_{\boldsymbol{\lambda}}$
σ_z^2	0.1	IPS-ALL	depth update related value of $\Sigma_{\boldsymbol{\lambda}}$
$\mathbf{b}_x \times \mathbf{b}_y$	7×7	ALL	patch size in pixel
$\mathcal{S}$	NCC	ALL	similarity measure

Table 3.7: Parameter settings for experiments.

of $m(1.25, 13, 0.1) = 449$ patch orientations that are evaluated at $31 \cdot 5 = 155$ 3D positions. In total $449 \cdot 155 = 69595$ hypotheses are evaluated for each pixel position. The computed results are illustrated in figure 3.27. The plots in the third row of the figure show a color-coded pixel-wise error. In order to visually exhibit smaller and larger errors at the same time, the visualized error is the absolute depth difference up to the power of 0.2, while the resulting values have been clamped to a range of $[0, 2]$. It can be seen that EPS constitutes an upper bound for the 3D surface reconstruction precision of all evaluated flavors of the Iterative Sweep. The high amount of evaluated hypotheses enables EPS to generate results up to the quantization precision of the image textures. In contrast, the presented IPS flavors only evaluate six to seven hypotheses for a pixel position during one iteration and the Monte Carlo based update sampling of IPS might provide some of the required hypotheses only after a huge amount of iterations. However, after a few iterations, the IPS results are already very close to the EPS reference as it can be seen by comparing the EPS accuracy

from table 3.8 with the iteration versus error plots of IPS in figure 3.29. A comparison of the IPS flavors shows the benefit of including surface orientation into the hypothesis representation. While the reconstruction of the plane is almost identical for IPS-DEP and IPS-DEPN(+SH), the quality of the sphere surface is much different. Especially, near the kink where the surface normals are almost orthogonal to the camera direction, the fronto parallel patches that are used with IPS-DEP are not adequate for a proper reconstruction. In addition, it can be seen that the supplementary *smoothness hypothesis* as it is used with IPS-DEPN+SH leads to a significantly smoother and more precise surface reconstruction.

	tolerance in millimeter					
	0.1	0.25	0.5	1.0	2.0	mean error
EPS	79.69	92.44	95.60	97.64	98.91	0.1384
IPS-DEP	36.96	60.86	76.10	88.43	95.72	0.4554
IPS-DEPN	62.42	87.07	94.42	97.31	98.86	0.1763
IPS-DEPN+SH	76.85	89.99	94.71	97.19	98.87	0.1533

Table 3.8: Integrity percentage with respect to some tolerance thresholds and mean absolute depth error in millimeter. Plots of the corresponding disparity errors can be found in figure 3.28. The values for IPS were obtained after 50 iterations.

The observations regarding the qualitative evaluation in figure 3.27 can also be confirmed by a numerical comparison with respect to the ground truth 3D data. Here, the fraction of depth values that are reconstructed with an absolute distance that is smaller than some threshold value is used as the quality criterion. It is referred to as the integrity of the result. In table 3.8, there is a listing of the major integrity percentages for the different sweeping procedures together with the mean absolute error. It can be seen that all variants reach almost the same integrity level for an error tolerance of 2.0 mm or higher. However, for tighter error thresholds there are significant differences. The surface approximation via oriented patches can be identified as a superior approach. While IPS-DEPN already exhibits an integrity of 62.42 percent for 0.1 mm tolerance, there is an significant additional gain for adding the *smoothness hypothesis*. Regarding the mean absolute error, figure 3.28 illustrates the relationship between inaccurate pixel correspondences and depth errors according to

$$\Delta\eta = \frac{\Delta z}{z^2} \cdot b \cdot f, \tag{3.44}$$

where b denotes the baseline, f the focal length, z the depth, Δz the depth error and $\Delta\eta$ the disparity error. Besides the increased precision, IPS-DEPN+SH also provides a faster convergence than IPS-DEPN as illustrated in figure 3.29. In this context, please note that all experiments have been conducted without multi-scale processing that would speed up the convergence even further as discussed in section 3.2.3.5.

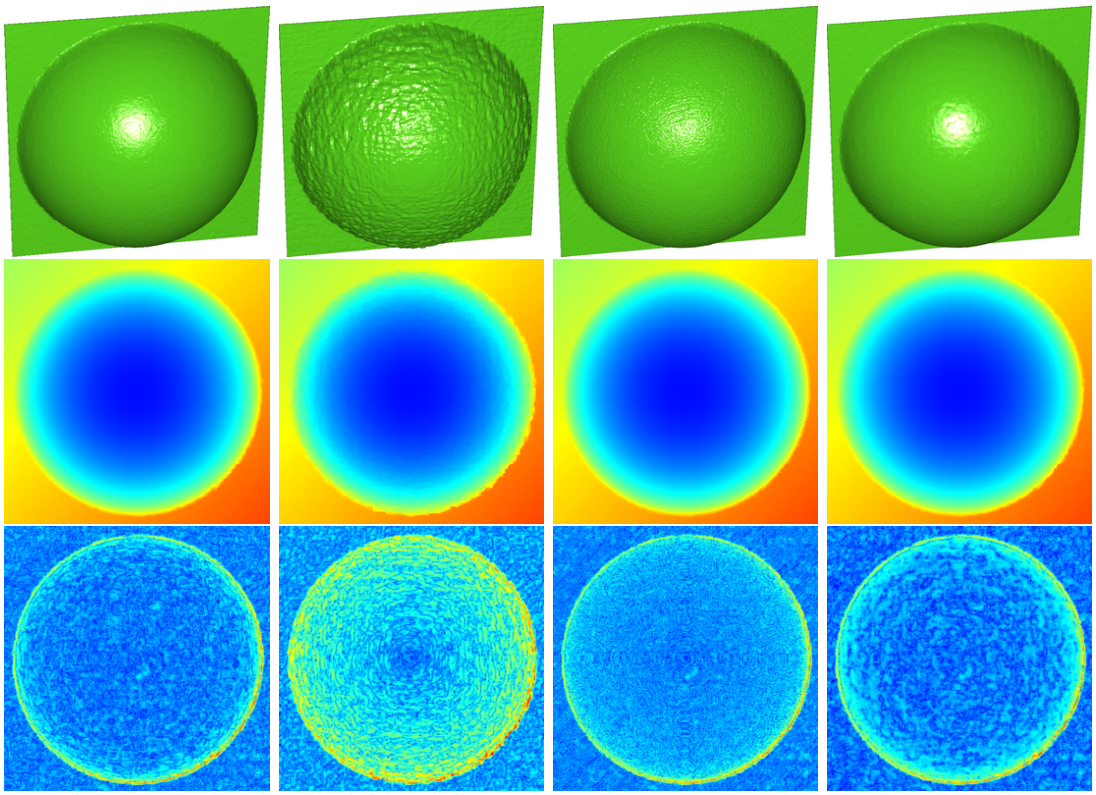


Figure 3.27: Computed results for different Patch-Sweep variants. The IPS results were obtained after 50 iterations. **From left to right:** EPS, IPS-DEP, IPS-DEPN and IPS-DEPN+SH. **First two rows:** Rendered depth mesh and color-coded depth map. **Third row:** Error plots with respect to ground truth.

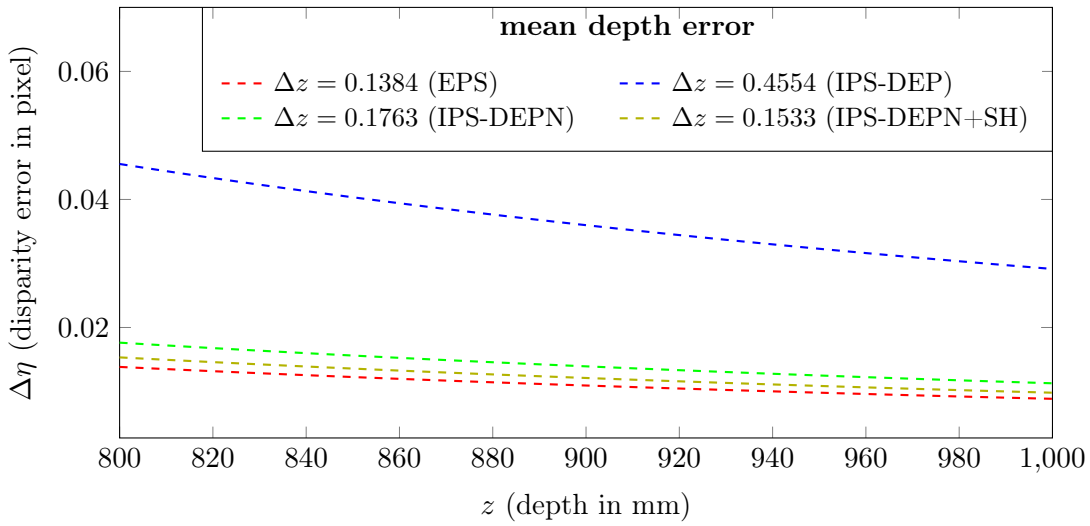


Figure 3.28: Relationship of depth errors and pixel correspondence inaccuracies for the synthetic camera parameters b=40 mm and f = 1600 pixel. The selected depth range matches the depth range of the synthetic 3D object. Each plot reflects the disparity error for this depth range with respect to the evaluated mean depth errors from table 3.8.

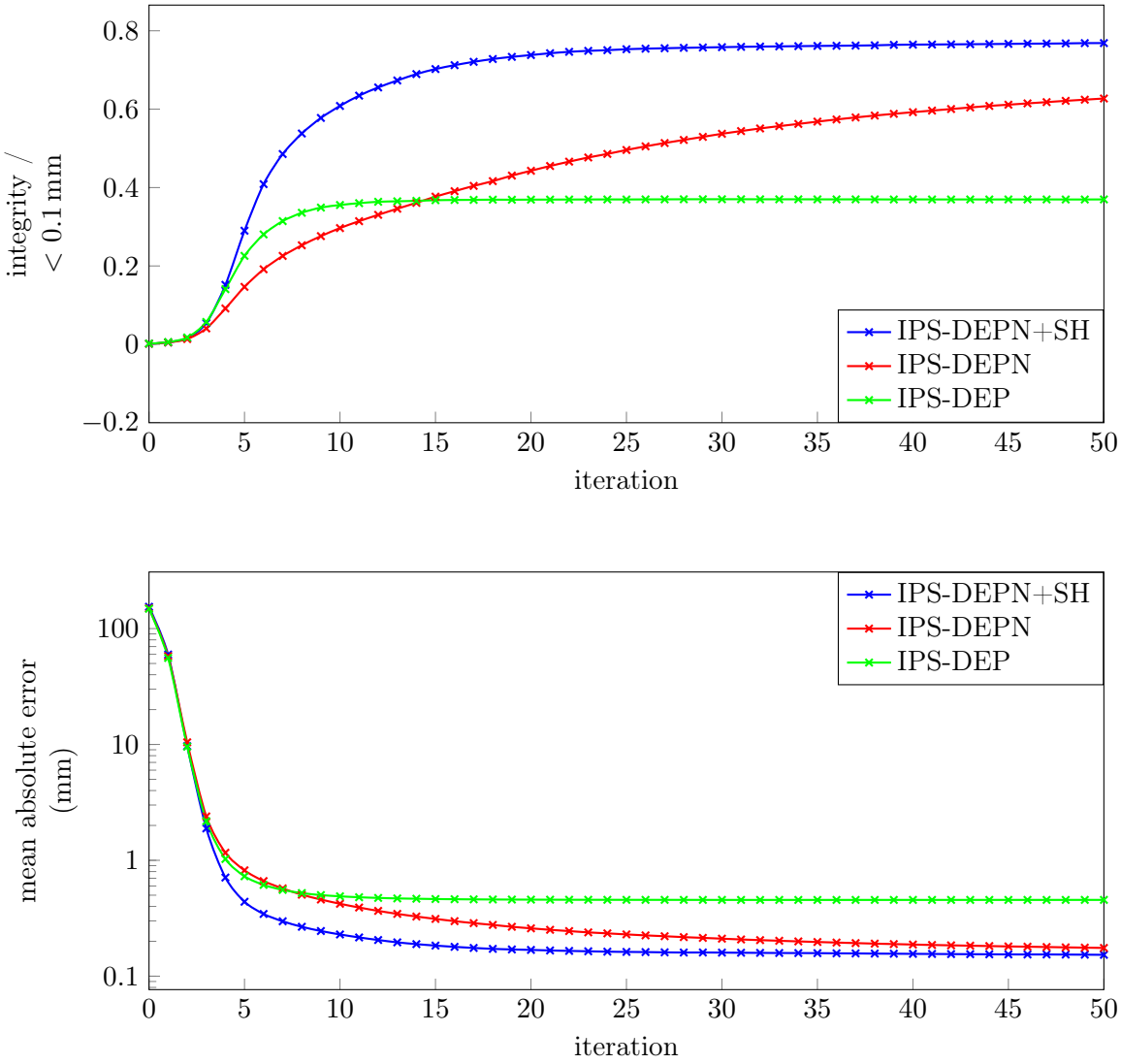


Figure 3.29: Top: Evolution of integrity for the Iterative Sweeping variants with a very tight threshold of 0.1 mm. **Bottom:** Evolution of the mean absolute error.

3.2.4.2 Stereo versus Trifocal

As outlined in sections 3.2.1 and 3.2.3.1, the structure of the Patch-Sweep algorithm allows for 3D processing on multiple views. This section is devoted to a qualitative comparison of the IPS results for stereo and trifocal input. The focus is on confirming that the algorithmic design as depicted in figure 3.16 accounts for an increased completeness in case additional images are available. Therefor, the Iterative Sweep is configured with a consistency threshold of T_c =10 mm. Exemplarily, results for stereo and trifocal processing on the `Oliver1` dataset are illustrated in figure 3.30. The three input views are shown on the top of the figure. The two rightmost views are used for stereo processing. On the bottom of the figure, the stereo result is shown on the left and the trifocal result on the right. As it could be expected from the view perspective of the cameras, the stereo result is degraded on the right side of the person's face. In contrast, the trifocal result exhibits an enhanced completeness due to the availability of an additional view. In consequence, the algorithmic concept for increased completeness with an additional input image could be empirically verified.

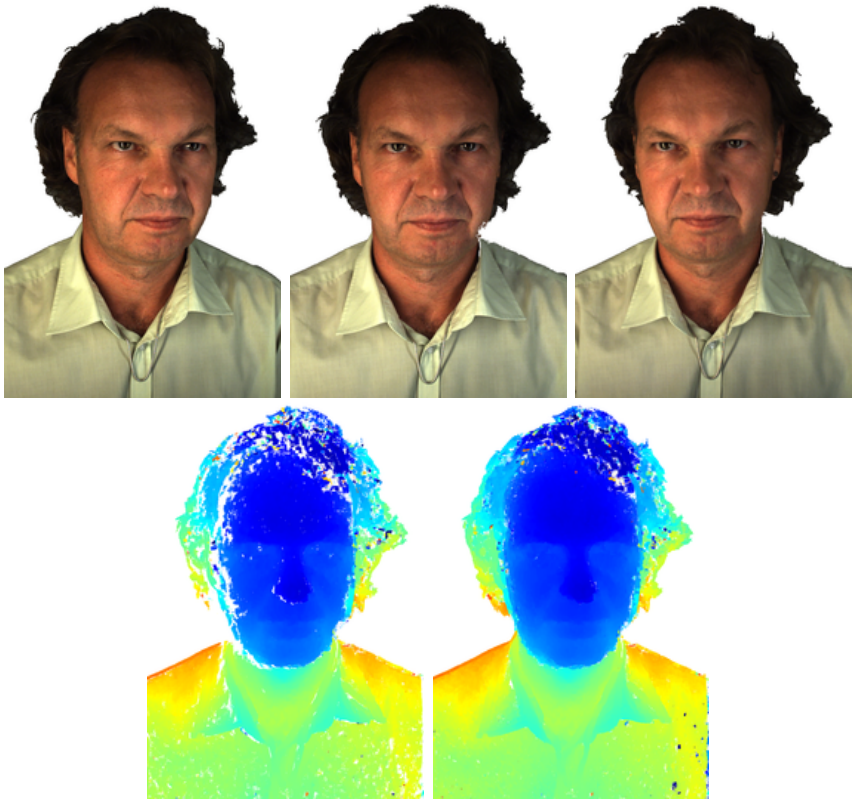


Figure 3.30: Comparison between stereo and trifocal camera configurations for the `Oliver1` dataset. **Top row:** Three input images. **Bottom row from left to right:** Result for stereo and for trifocal processing. The white areas indicate that the results for this positions did not pass the consistency check.

3.2.4.3 State-of-the-art Real-time Stereo Comparison

The purpose of the following experiments is the comparison with state-of-the-art real-time disparity estimation algorithms. There are three evaluation targets: First, a comparison of quality with respect to video communication datasets. Second, a performance evaluation

in order to classify the efficiency of the IPS. And third, an evaluation of the scalability, i.e. the performance gain for adding additional processing units. The selected state-of-the-art reference algorithms are Line-Wise Hybrid Recursive Matching (L-HRM) [RZK11] and Semi-Global Matching (SGM) [Hir05]. The choice of those algorithms is motivated by two criteria. On the one hand, both algorithms are designed for real-time processing. L-HRM is dedicated as a CPU based reference and SGM serves as a GPU based reference. While the initial SGM was designed for CPU, there are efficient GPU implementations of SGM in literature. Unfortunately, the reported performance cannot be directly compared to the presented work, since different hardware was used. Only a rough impression about the performance range can be given. For example, the authors of [Mic+13] achieved with SGM 11.7 fps for VGA input resolution and 64 disparity levels on a NVIDIA Geforce GTX 480 graphics card. On the other hand, both algorithms were robustly applied on production level within the real-time domain. Especially, L-HRM was used in various practical real-time applications [Rie+12a, Rie+12b], where the focus is on view synthesis of video streams as it is the case in an eye contact preserving video communication scenario.

relevance	**parameter**	**value**	**description**
IPS-DI	N_{iter}	1	fixed number of iterations
IPS-DI	r_s	4	number of random hypotheses
IPS-DI	σ_r^2	10	diagonal values of $\Sigma_{\mathcal{N}_r^4}$
IPS-DI	σ_η^2	15	disparity update related values of $\Sigma_{\boldsymbol{\lambda}}$
IPS-DI	L	2	sweep levels
IPS-DI	s_{lev}	0.3	sweep level scale factor
IPS-DI / L-HRM	T_c	3	consistency threshold (pixel)
IPS-DI / L-HRM	$\mathcal{S}$	NCC	similarity measure
IPS-DI / L-HRM	$\mathbf{b}_x \times \mathbf{b}_y$	16×16	block size (pixel)

Table 3.9: L-HRM and IPS-DI parameter settings for stereo evaluation.

In order to enable for a meaningful comparison, the Iterative Sweep is configured for a Monte Carlo based disparity update (IPS-DI) as discussed in section 3.2.3.2 and 3.2.3.6. The hardware for performance evaluation is chosen as listed in appendix A. While the hardware setup exhibits a multi-view system, the evaluation concentrates on the processing of a single stereo input with a fixed set of parameters for all datasets. In order to measure the scalability, the allowed hardware consumption is varied between 1 to 32 CPU threads for L-HRM and 1 to 7 graphics cards for the Iterative Sweep. One of the pursued targets is to evaluate the peak performance of each algorithm on the available hardware. Here, a CUDA based multi-GPU IPS-DI implementation is compared with a highly SSE optimized multi-threaded C++ implementation of L-HRM [RZK11]. A comparison with L-HRM on GPU is out of scope for this analysis as L-HRM can only be parallelized on the level of image rows and an efficient GPU implementation with full hardware utilization is not possible. Since a GPU based implementation of SGM like [Mic+13] was not available to the author, the performance of SGM could only be compared based on the literature. In contrast, for

quality comparison, the single threaded SGM implementation from OpenCV [Its15] that exhibits a runtime of several seconds for full HD stereo input was used. The IPS-DI and L-HRM parameter settings for all experiments were fixated according to table 3.9 and the SGM settings were selected as suggested by the OpenCV documentation [Its15].

The two resolutions UHD (3840 x 2160) and HD (1920x1080) were selected for the input stereo streams. For this purpose appropriate regions of interest of the high resolution datasets `SaschaHR`, `RonnyHR`, `HannesHR` and `JohannesHR` were used to serve as UHD input. For HD processing, the same input was down-scaled prior to stereo estimation.

	IPS-DI	SGM	L-HRM
`HannesHR`	93.158	90.173	90.277
`RonnyHR`	94.630	94.308	95.714
`SaschaHR`	95.200	92.929	95.536
`JohannesHR`	95.138	91.752	94.379

Table 3.10: Average completeness percentages for 500 frames of each listed dataset.

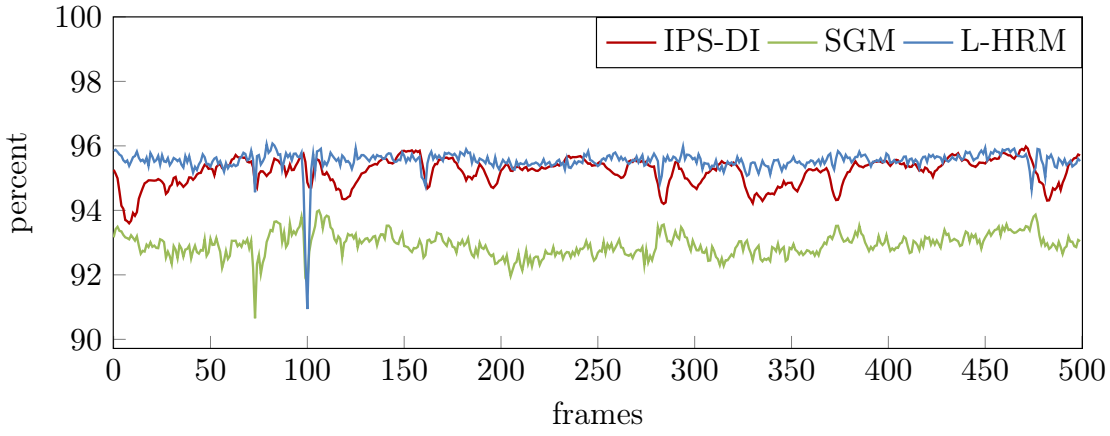


Figure 3.31: Frame wise evaluation of the stereo estimation completeness for 500 frames of the `SaschaHR` dataset.

A qualitative comparison of the results on the `SaschaHR` dataset for all three algorithms is provided in figure 3.32. It can be seen that all results exhibit a comparable quality for still images. However, when comparing the resulting disparity video sequences, the IPS-DI results show significantly less flickering artifacts than those from L-HRM and SGM. The consistency of the results was evaluated numerically. For IPS-DI and L-HRM, a consistency check according to section 3.2.3.1 was performed with a threshold of $T_c = 3\,\text{pixel}$. The same threshold was set for the pseudo consistency check that is used within the OpenCV implementation of SGM. An overview to the mean completeness values for 500 frames of the `SaschaHR`, `RonnyHR`, `HannesHR` and `JohannesHR` datasets is provided in table 3.10. Exemplarily, the frame wise completeness for the `SaschaHR` dataset is illustrated in figure 3.31. It can be seen that IPS-DI can match or even outperform the completeness of SGM and L-HRM on the evaluated datasets.

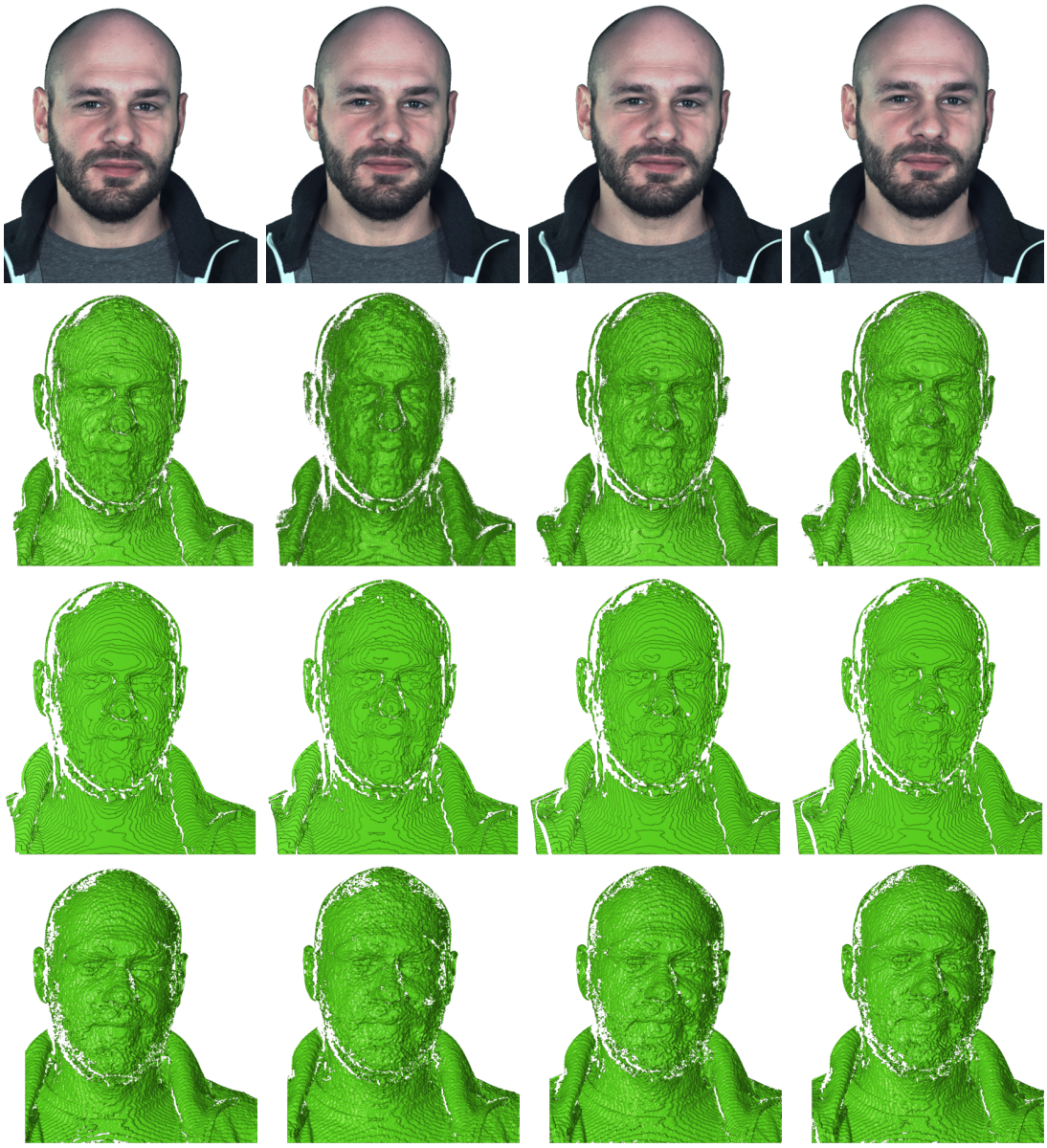


Figure 3.32: Qualitative evaluation for IPS-DI, L-HRM and SGM. **From top to bottom:** Left input texture, IPS-DI results, L-HRM results, SGM results.

The computation times for L-HRM and IPS-DI on HD and UHD resolution are listed in table 3.11. There are two columns for each resolution in order to compare timings for dense and 4×4 grid processing. As grid processing leads to sparse results, it is commonly used in combination with filter post-processing [Rie+12b]. Each row of the table shows the computation times for a certain level of parallelization of IPS-DI and L-HRM. The GPU based IPS-DI exhibits a significant performance advantage even if L-HRM is parallelized on 32 threads. From a practical perspective the possible real-time throughput of stereo pairs for a given hardware configuration is of interest. Figure 3.33 shows a comparison of stereo image throughput at 30 fps for IPS-DI and L-HRM with HD resolution and with respect to full CPU resource consumption of the PC as listed in appendix A. It can be seen that IPS-DI is able to process 4.15 HD stereo pairs on a 4×4 grid on a single GPU while L-HRM requires 32 CPU threads to achieve a throughput of 0.87 HD stereo pairs.

Please note, that this comparison does not intent to compare the algorithmic performance of IPS-DI and L-HRM in terms of arithmetic operations that are required to compute a result. Instead, the main target is to illustrate the degree of hardware saturation of both algorithms for the targeted video communication application. Regarding the illustrated stereo image throughput the main advantage of IPS-DI is based on the parallel algorithmic design that allows for an very efficient implementation on graphics hardware. From the perspective of arithmetic operations, there is no significant difference between L-HRM and IPS-DI. Although the timing of SGM could not be directly compared, a rough estimate based on the performance of the different graphics hardware and the numbers from literature can be made. The NVIDIA Titan X that is used for IPS-DI exhibits 4.9 times more GFLOPS than the NVIDIA GTX 480 that was used by [Mic+13]. Extrapolating the reported runtime from [Mic+13], these numbers lead to a computation time of approximately 118 ms on HD resolution.

Finally, a comparison of the algorithmic scalability between L-HRM and IPS-DI is conducted. The figure 3.34 shows speedup graphs with respect to the consumed hardware. The y-axis of the graph shows the fraction of achieved speedup and the number of processing units (PU). In the ideal case of a linear scaling, all numbers would be constantly one. In contrast, a value of 0.5 would indicate that half of the applied hardware did not contribute to parallelization. For dense processing with IPS-DI, the slopes of the graphs for HD and UHD both show an almost linear but shallow drop of achieved speedup with increasing number of PUs. For 4×4 grid processing, all slopes show a much steeper drop down than for dense processing. With a block size of 16, almost all texture values of the stereo input must be read at least once. However, due to processing on a 4×4 grid the cache cannot hide the bandwidth limitation in contrast to dense processing where most values have to be accessed many times. In consequence, additional processing units do not lead to an efficient speedup neither for CPU nor for GPU parallelization.

	Threads/	HD		UHD	
Algorithm	GPUs	dense	4×4	dense	4×4
IPS	1	87.8 ms	8.0 ms	347.8 ms	30.0 ms
IPS	2	46.0 ms	5.5 ms	187.5 ms	20.0 ms
IPS	3	32.0 ms	5.0 ms	133.5 ms	16.0 ms
IPS	4	26.4 ms	4.5 ms	100.6 ms	14.8 ms
IPS	5	22.4 ms	5.0 ms	86.7 ms	14.7 ms
IPS	6	20.0 ms	6.0 ms	76.0 ms	14.0 ms
IPS	7	18.2 ms	6.1 ms	67.4 ms	13.8 ms
L-HRM	1	7565.2 ms	519.4 ms	29363.8 ms	2138.3 ms
L-HRM	2	4078.9 ms	298.5 ms	16225.4 ms	1234.9 ms
L-HRM	4	2217.1 ms	243.9 ms	8934.6 ms	828.8 ms
L-HRM	8	1261.3 ms	130.6 ms	4663.7 ms	468.2 ms
L-HRM	16	853.9 ms	70.6 ms	3131.8 ms	321.9 ms
L-HRM	32	579.2 ms	38.0 ms	1881.8 ms	228.4 ms

Table 3.11: Runtime for dense and 4×4 grid processing on UHD and HD resolution.

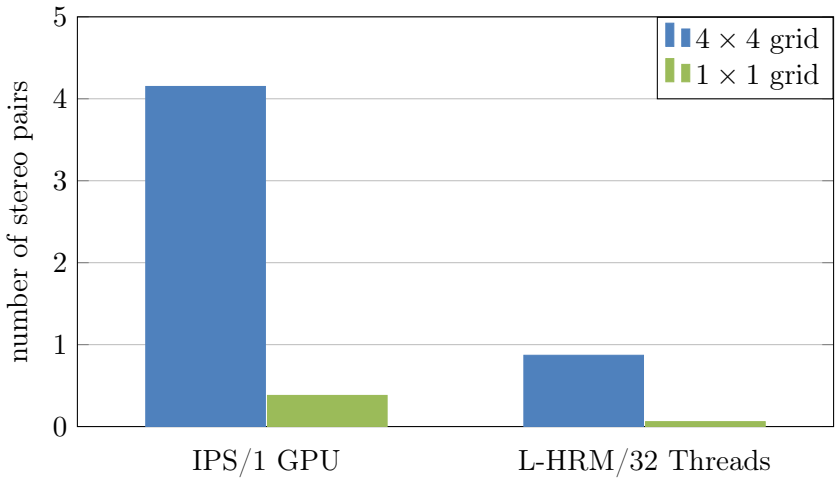


Figure 3.33: Number of HD stereo pairs that can be processed with 30 fps on a single PC.

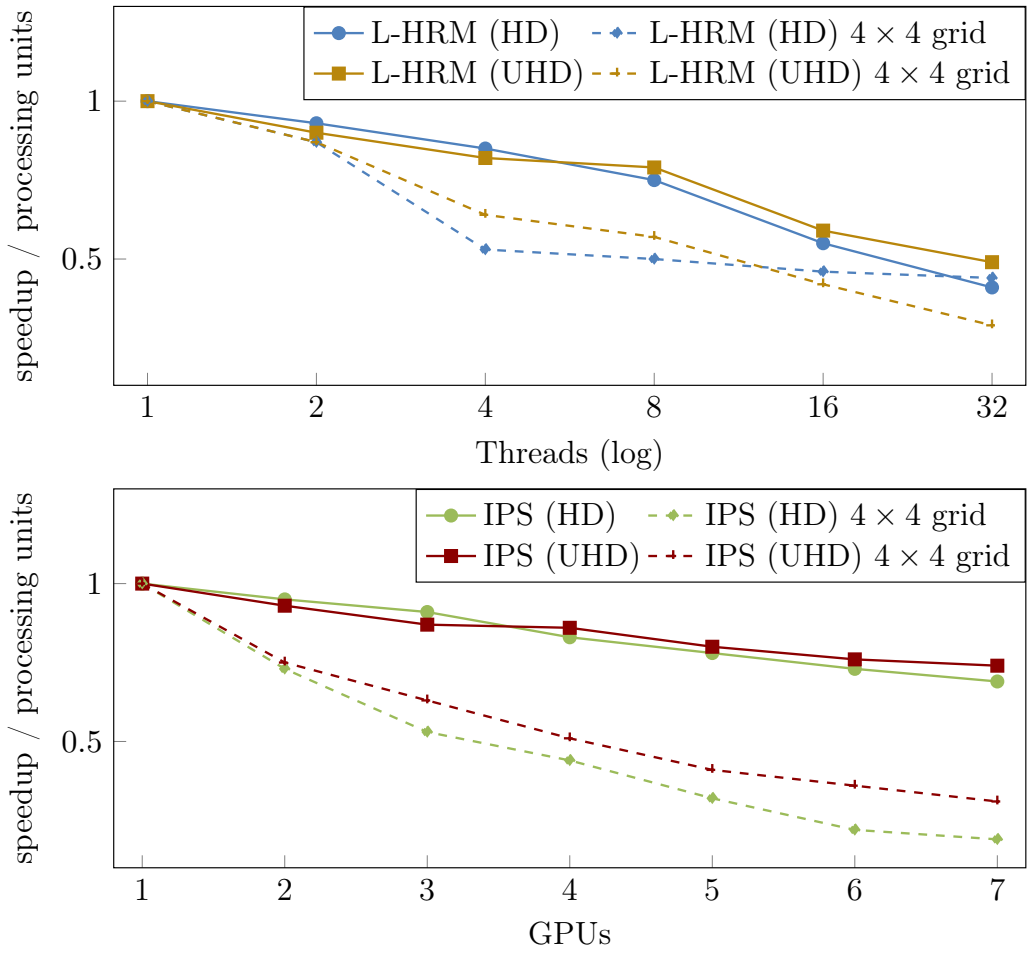


Figure 3.34: Comparison of achieved speedup with respect to number of processing unit.

3.2.4.4 Combination of Visual Hull and Patch-Sweep

In this section, a possible combination of the Patch-Sweep algorithm and the depth data from IBVH computation is discussed. The proposed approach leads to a significant improvement of the completeness of the results, especially in case of less structure on the clothes of the conferees. Inherently, the Visual Hull represents the outer bound of an object according to the available silhouettes. In consequence, a Visual Hull depth map provides a lower bound for a valid depth search range as it was outlined in section 3.2.2. According to table 3.2, there are little quantization artifacts for IBVH results at QHD and QQHD resolution. Hence, IBVH depth maps with QQHD resolution are used in order to maintain a good cost-value-ratio in terms of quality versus computational complexity. These depth maps are upscaled to the desired input resolution and used in conjunction with the Iterative Sweep as follows. According to equation (3.13), a pixel-wise sweeping range is defined and EPS hypotheses lists are generated for each pixel based on the parameter set in table 3.12. However, these additional hypotheses are not used for Exhaustive Sweeping, but they are used to extend the *basic* IPS hypotheses lists of equation (3.23). In this way, the iterative procedure is guided by additional hypotheses that are sampled within a small depth range where the object boundary is presumably located. Based on the extended hypotheses list, IPS is executed regularly with the trade-off configuration as listed in table 3.5 and a consistency check is performed subsequently. For each pixel position with an inconsistent result, the consistency of the IBVH depth for the left and right image is evaluated. If it passes the consistency check, the inconsistent result is replaced with the IBVH depth. An overview to the average completeness improvements is provided in table 3.13. It can be seen that the achieved completeness values outperform the previous result for all datasets. In particular, the challenging `Sylvain` and `Marcus` datasets can greatly benefit from the shape guided handling of the homogeneous image regions. Exemplarily, figure 3.35 provides a frame wise comparison for the `Sylvain` dataset as it contains the most challenging content among the six datasets. Please note that not only the average completeness was improved, but every single frame exhibits a superior completeness in case of the proposed combination with IBVH. A qualitative evaluation of the result improvements for the `Sylvain` dataset is illustrated in figure 3.36.

parameter	value	description
z_r	20	initial begin of depth range (mm)
N_D	5	depth discretization steps
N_R	0	coarse to fine steps
L_l	1	lines of latitude

Table 3.12: EPS parameter settings for the generation of hypotheses that are based on the IBVH depth map input. The listed parameters enables for five additional hypotheses.

	EPS	best	trade-off	trade-off + IBVH
David	93.87	94.07	93.79	95.66
Sylvain	53.81	67.75	66.82	89.97
Marcus	56.23	79.09	78.16	93.45
Paul	87.82	92.59	92.29	94.48
Niklas	88.09	93.30	93.02	95.13
Oliver2	88.18	92.44	92.16	94.32

Table 3.13: Dataset wise average completeness values for results with and without depth input from IBVH computation. In order to facilitate the comparison with previous results, the columns for EPS, best and trade-off results were copied from table 3.6.

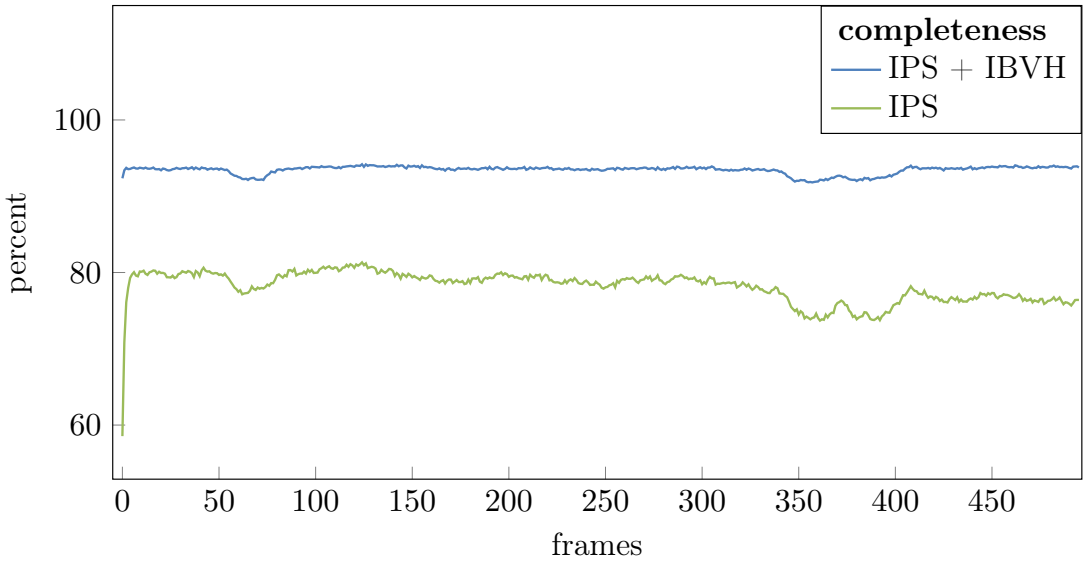


Figure 3.35: Comparison of IPS results for the Marcus dataset with and without additional IBVH input. For both results, the trade-off configuration as listed in table 3.5 was used.

Figure 3.36: Results for the Sylvain dataset that illustrate the different completenesses for homogeneous image regions. **From left to right:** Left input frame, IPS results, IPS with additional depth input from IBVH. The white areas indicate inconsistent depth values.

3.2.4.5 State-of-the-art Multi-view 3D Comparison

In order to render the eye contact view as discussed in chapter 5, composed 3D data from multiple cameras is required. For this purpose, the depth results of the Patch-Sweep algorithm are combined in terms of a visibility-driven real-time 3D fusion algorithm [Ebe+14]. In the following, the resulting patch group representations are qualitatively evaluated with the results of a popular state-of-the-art 3D reconstruction work-flow that consists of the Patch-based Multi-view Stereo Software (PMVS2) from Furukawa *et al.* [Fur, FP07, FP10] and a constitutive Poisson reconstruction of the object surface [Kaz, KBH06, KH13] and vertex coloring [CT12]. The Patch-Sweep results were computed with the *trade-off configuration* that is listed in section 3.2.3.6. The settings for PMVS2 and the Poisson reconstruction were carefully tuned for high quality. A parameter listing is provided in table 3.14. A comprehensive explanation of possible parameter settings can be found in the software documentations [Fur, Kaz]. The comparison is conducted for all multi-view video communication datasets with sixteen views that are listed in appendix B. To enable a qualitative comparison, the results for one frame of each dataset are exemplarily illustrated in figure 3.37 and figure 3.38. It can be seen that the fused Patch-Sweep results exhibit a comparable quality and completeness as with the PMVS2 and Poisson reconstruction approach. In particular, the pure Patch-Sweep results on the left of figure 3.38 are much denser than the PMVS2 output on the left of figure 3.37. At the same time, in contrast to the real-time computation of the Patch-Sweep algorithm, for a single frame, the computation times for PMVS2 and Poisson reconstruction were in the range of 20 to 30 minutes on the computer that is listed in appendix A.2.

relevance	**parameter**	**value**	**description**
PMVS2	$level$	0	the input images are scaled with factor 2^{-level}
PMVS2	$csize$	2	for every $csize \times csize$ pixel square regions, the software tries to extract at least one patch
PMVS2	$threshold$	0.4	acceptance threshold for patch reconstruction
PMVS2	$wsize$	21	$wsize \times wsize$ pixel are sampled for photometric consistency
PMVS2	$minImageNum$	2	each 3D point must be visible in at least $minImageNum$ images
PMVS2	$maxAngle$	1	minimal angle between two cameras to enable the reconstruction of 3D points
Poisson	$depth$	10	octree depth
Poisson	$color$	16	balance factor for color estimates
Poisson	$trim$	8	density based mesh trimming value

Table 3.14: Parameter settings for PMVS2 and Poisson surface reconstruction. For PMVS2 the visibility mask was configured in order to match the stereo configurations that where used for Patch-Sweep processing.

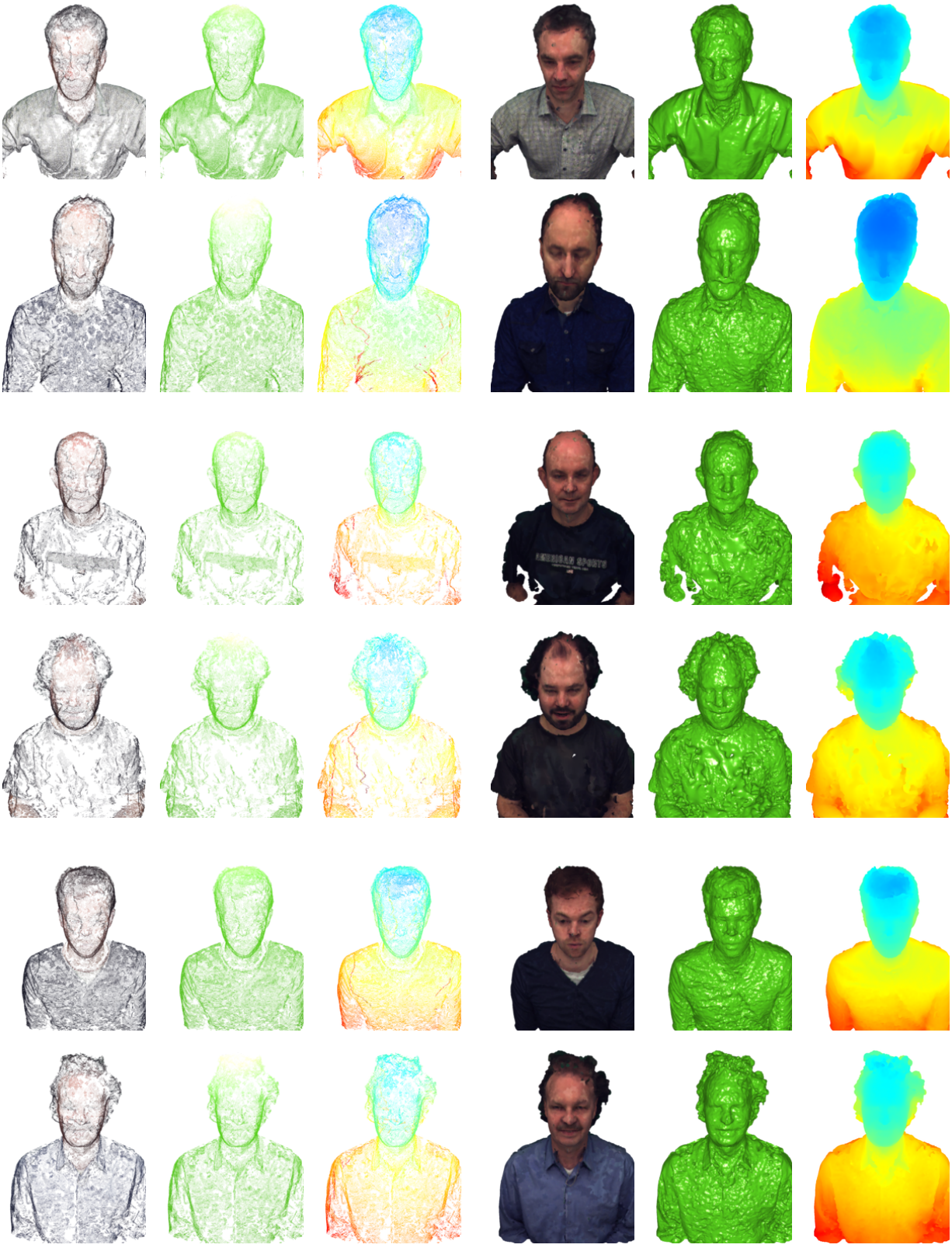


Figure 3.37: Results for PMVS2 and Poisson surface reconstruction. The three leftmost columns show the oriented point cloud output from PMVS2 and the three rightmost columns are the results after Poisson surface reconstruction. For each result, the colored 3D data, the uncolored 3D data and the depth map of the selected perspective is presented.

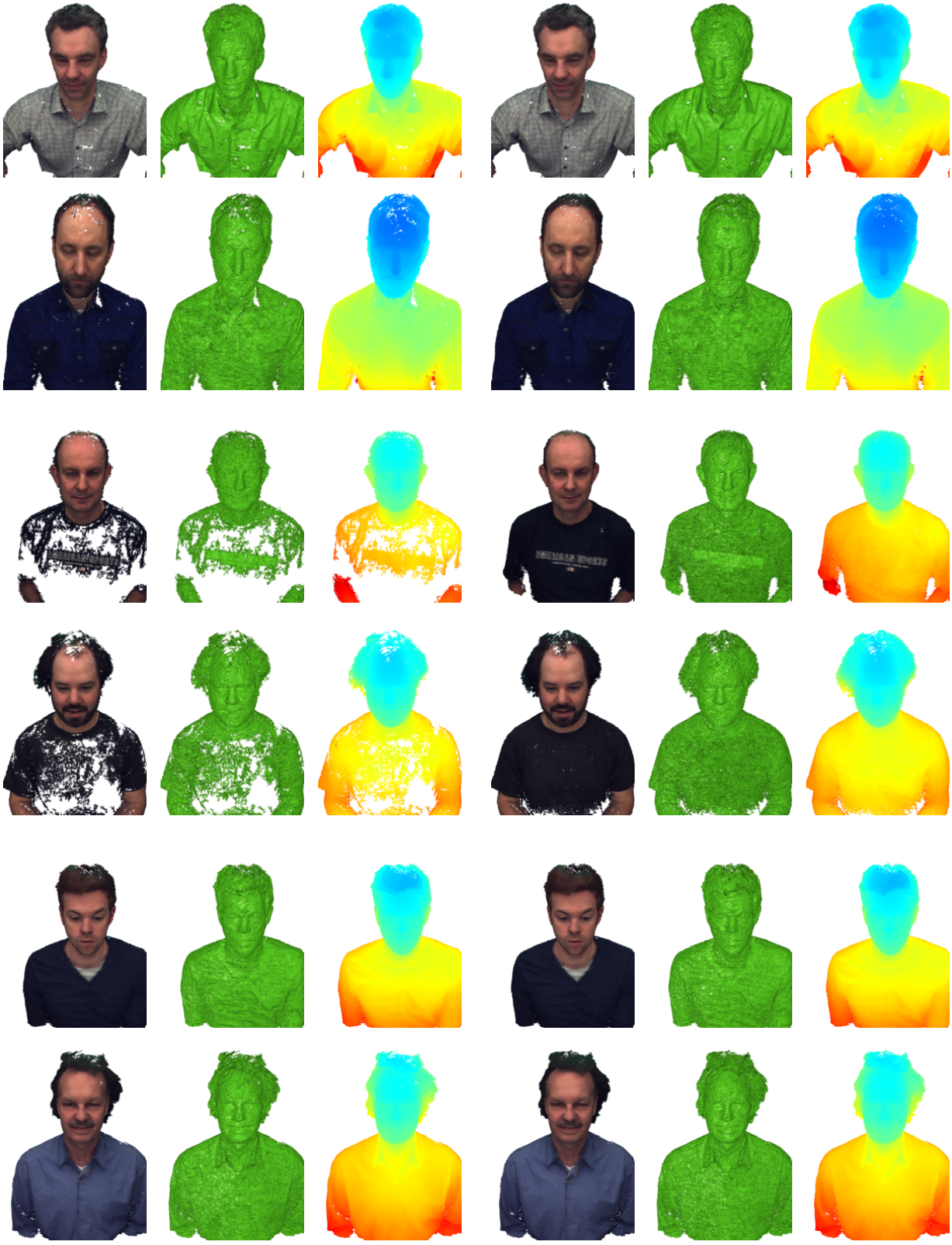


Figure 3.38: Multi-view fusion of Patch-Sweep results. The three leftmost columns show the results for pure Patch-Sweep fusion and the three rightmost columns are the fusions results for the proposed combination of Patch-Sweep and IBVH. For each result, the colored patch groups representation, the uncolored patch group representation and the depth map of the selected perspective is presented.

3.3 Chapter Summary

In this chapter, a novel IBVH variant and an iterative patch-based stereo estimation framework were introduced. The application-inherent real-time constraints were addressed with a strong focus on computationally efficient and massively parallel algorithms that can be executed on many-core systems like modern GPUs. It was shown that the proposed parallel IBVH is able to process 16 HD input streams on a single state-of-the-art graphics card within real-time. Beside parallel processing, the presented pixel preselection could leverage an additional average speedup of up to 45 percent on the evaluated datasets. For objects that are not completely visible in all silhouette images, a novel image border extension was proposed that is able to prevent a cutoff of the IBVH result. The presented patch-based iterative stereo estimation framework enables for a customization of the representations, update and propagation of hypotheses. Therefore, depending on the application requirements, the framework allows for the compilation of a domain specific algorithmic flavor. With a focus on video communication, different algorithmic combinations were proposed. Their efficiency and the quality of results were successfully evaluated based on synthetic data and real-world video communication datasets. Depending on the selected hypotheses representation, the evaluation on the synthetic data exhibit the potential for a very high precision with respect to the ground truth. The comparison of results for the video communication datasets with state-of-the-art stereo and multi-view algorithms shows a similar performance regarding the quality of results, while there is a significant advantage of the proposed approach regarding the computational performance. In addition, in case of matching ambiguities due to less structured image regions, the iterative propagation and update of hypotheses leads to a more complete result compared to an exhaustive sweeping procedure. A further improvement regarding homogeneous image regions was accomplished by the inclusion of IBVH depth data for the generation of additional hypotheses. Therefor, the proposed combination with IBVH constitutes a convenient alternative to the integration of depth sensors.

4. Continuous Photometric Alignment

A careful adjustment of image colors is advantageous and vital for many multi-view computer vision algorithms. Among other, computational stereo and a constitutive view synthesis can greatly benefit from high precision photometric alignment. In exhaustive studies, it was reported that the disparity estimation results for a stereo image pair recorded with the very same camera, i.e. identical photometric settings tend to be superior to those taken with different color adjustments [HS09, HS07]. Although most state-of-the-art cameras support automatic white balancing, the results are not reliable if consistent color settings are desired. First, depending on the camera placement, the automatic white balancing might lead to diverging settings among cameras if it is performed from different perspectives with potentially unequal lighting conditions and scene colors. Second, since automatic white balancing is performed independently for each camera, there is no constraint that enforces matching colorimetric settings even if the cameras see exactly the same part of a scene. But also with a careful manual balancing of all camera settings, it is difficult and time-consuming to adapt and match their photometric properties. While a sophisticated color chart based manual photometric adjustment is possible under lab conditions, there are many applications that do not allow intricated user interaction. Particularly, user centric on-line stereo and multi-view systems, like eye contact preserving video-communication solutions, require a fully automatic color registration workflow. Regarding the real-time 3D analysis in chapter 3, where the computational load is of significant interest, matching photometric properties among all cameras can greatly help to reduce the computational complexity. Instead of specialized and expensive similarity measures that compensate for different radiometric conditions, much simpler similarity measures and smaller window or patch sizes could be used while results of comparable quality can be expected [HS09, HS07, HLL08, WYD07]. At the same time, consistent color settings allow for a seamless view synthesis from different input cameras without complex texture blending.

The photometric alignment is conducted with respect to the color settings of a predefined reference camera. The colorimetric properties of this camera can be chosen manually, via one shot or continuous automatic white balancing or any other user favored method. The algorithm presented in this section cares about keeping the color settings of all other cameras exactly synchronized. For this purpose, each camera is pairwise adjusted with respect to the defined reference camera. In consequence, without loss of generality, the following algorithmic description focuses on a single stereo configuration.

The main contribution of this chapter is a novel depth driven algorithm for high accuracy combined geometric and photometric stereo image registration. The goal is to optimize photometric camera settings with respect to optimal depth estimation results. The algorithm is capable of a continuous fully automatic on-line adjustment of colorimetric camera settings and of the electronic off-line fine-tuning of photometric properties for recorded stereo sequences. The registration process is formulated in terms of an alternating energy minimization procedure, where the geometric and photometric registration energies are consistently incorporated into the same continuous energy functional. On the one hand, the alternating optimization consist of an iterative step for the geometric registration, while the photometric parameters are fixed. On the other hand, the photometric parameters are optimized while the geometric registration remains fixed. The formulation of the energy functional allows for the application of the powerful machinery of the variational calculus in order to optimize the geometric image registration for a state of minimal energy. Based on this solution, the photometric parameters are optimized via a gradient descent approach. Both steps are repeated until convergence. As a depth-based geometric image registration and the photometric registration is pursued concurrently, the quality of photometric registration is directly related to the performance of depth estimation and vice versa. Therefore, the approach is perfectly suited to enhance the outcome of stereo and multi-view algorithms and it improves the visual experience of a constitutive view-synthesis. Regarding the implementation perspective, the presented registration method is designed with focus on parallelizability which allows for an efficient real-time implementation on graphics hardware. In a typical application scenario for video communication, the registration procedure is performed in advance. In case a change of lighting conditions is expected during communication the registration is performed concurrently in background with a reduced frame rate. For both options, during the video communication session, there is no or only very little computational overhead for applying the registration procedure.

In the following, section 4.1 covers the general formulation of the depth driven registration approach in terms of energy minimization. Subsequently, section 4.2 and 4.3 introduce the continuous energy functional and the steps for geometric and photometric image registration respectively. In section 4.4, the fusion of both steps into an alternating iterative minimization scheme is described. Section 4.5 illustrates the successful application of the presented technique on different datasets. Parts of this chapter have already been published in [WFE11, 13b].

4.1 Image Registration in Terms of Energy Minimization

From a general point of view, the proposed geometric and photometric stereo registration algorithm is defined in terms of an energy minimization problem that is composed as

$$\mathcal{E}(u, \mathrm{T}) := \mathcal{E}_S(u) + \omega\, \mathcal{E}_D(u, \mathrm{T}), \tag{4.1}$$

where the total energy $\mathcal{E}$ consists of a *smoothness* term $\mathcal{E}_S$ and a *data* term $\mathcal{E}_D$ weighted by a scalar $\omega > 0$. The variables u and T denote the geometric and photometric registration

functions respectively. For simplicity, it is assumed that the stereo image pair is in a rectified state. This can be accomplished by image rectification transformation based on camera calibration parameters or state-of-the-art on-line camera adjustment techniques, e.g. [Zil+10]. Therefore, neglecting photometric registration and supposing color constancy, for a stereo image pair I_F, I_E an optimal geometric registration $u : \Omega \to \mathbb{R}$ has to fulfill

$$I_F(\mathbf{x}) = I_E(x + u(\mathbf{x}), y), \tag{4.2}$$

where $\mathbf{x} = (x, y)^T$ are Cartesian coordinates in the image domain $\Omega \subset \mathbb{R}^2$, and an image I is considered as a mapping $I : \Omega \to \mathbb{R}^m$, $I(\mathbf{x}) := (I^{c_1}(\mathbf{x}), \ldots, I^{c_m}(\mathbf{x}))$ from Ω to a $m-$dimensional color space. Finally, the photometric registration $\mathrm{T} : \mathbb{R}^m \times \mathbb{R}^k \to \mathbb{R}^m$, with $\mathrm{T}(I(\mathbf{x}), \mathbf{p}) = (\mathrm{T}^{c_1}(I(\mathbf{x}), \mathbf{p}), \ldots, \mathrm{T}^{c_m}(I(\mathbf{x}), \mathbf{p}))$ is defined as a color transformation mapping with a $k-$dimensional parameter vector $\mathbf{p} = (p_1, \ldots, p_k)$. Rewriting equation (4.2) with respect to photometric registration leads to the very basic requirement for a combined geometric and photometric stereo image registration mapping:

$$I_F(\mathbf{x}) = \mathrm{T}(I_E(x + u(\mathbf{x}), y), \mathbf{p}). \tag{4.3}$$

In the following, the concrete continuous energy formulation according to the form of equation (4.1) is composed. It takes equations (4.2) and (4.3) into consideration and allows for an elegant, consistent and effective alternating estimation of the unknown geometric registration function u and the parameter vector $\mathbf{p}$, for an arbitrary color transformation mapping.

4.2 Globally Optimal Geometric Image Registration

The choice of an energy functional that allows for a globally optimal solution of the geometric image registration mapping u can be motivated based on the general discussion in section 4.1. Disregarding the photometric registration, in compliance with equation (4.2), the commonly used absolute differences

$$\mathrm{AD}(u, \mathbf{x}, I_F, I_E) = \sum_{i=c_1}^{c_m} \left| I_F^i(\mathbf{x}) - I_E^i(x + u(\mathbf{x}), y) \right| \tag{4.4}$$

can be applied as a pixel-wise measure for the image registration quality. Consequently, the integral of the absolute differences over the image domain is employed as the *data* term of the energy functional. For regularization, i.e. as a *smoothness* term, the total variation of u is used, which is a convex function that allows sharp discontinuities. In combination, the minimization task with respect to the unknown geometric registration function can be expressed in terms of the variational problem

$$\min_u \left\{ \int_\Omega |\nabla u(\mathbf{x})| + \omega \, \mathrm{AD}(u, \mathbf{x}, I_F, I_E) d\mathbf{x} \right\}. \tag{4.5}$$

A stationary point for this variational problem could be obtained by solving the corresponding Euler-Lagrange equations with a simple gradient descent approach. However, since the *data* term of the functional cannot be expected to be convex, a stationary point of the functional might not be the globally optimal solution of the variational problem. Additionally, gradient descent iteration approaches suffer from a low rate of convergence.

Therefore, the solution proposed in [Poc+08] is applied to solve the variational problem (4.5). The authors pursue a convexification of the original expression via functional lifting. This allows the computation of a globally optimal solution via an efficient primal-dual proximal point iteration algorithm. Moreover, the algorithm is well suited for parallelization on graphics hardware, which enables for real-time computation. Especially, considering a video sequence with little changes from one frame to another, the algorithm converges after a few iterations, because the primal and dual variables of the previous frame can be reused for the initialization of the iteration process for subsequent frame.

4.3 Depth Driven Photometric Image Registration

In the following, the algorithmic steps for a pure photometric registration are discussed while assuming that u is already available. As stated in section 4.1, the optimization is subject to a general application specific, parametric color transformation function T with the parameter vector $\mathbf{p}$ that act as the color distortion model. Hence, the energy functional is formulated with respect to the considerations of equation (4.3) and a generic iterative minimization scheme for arbitrary color distortion models is provided. An application specific example for a concrete color transformation mapping and the resulting parameter update is presented in conjunction with the experiments in section 4.5.

In order to allow for a combined photometric and geometric registration, a major requirement is a consistent formulation with respect to the geometric registration process in section 4.2. The insertion of the color mapping T into the formulation of the minimization task of equation (4.5) leads to the optimization problem

$$\min_{\mathbf{p}} \left\{ \int_{\Omega} |\nabla u(\mathbf{x})| + \omega \mathrm{AD}(u, \mathbf{x}, I_F, \mathrm{T}(I_E, \mathbf{p})) d\mathbf{x} \right\}. \tag{4.6}$$

Because of preconditioning factors such as auto white balancing or similar default camera settings, it can be expected that the identity mapping is already close to the solution state. Therefore, local minima are unlikely and a steepest descent approach can be applied for optimizing. Since $\mathrm{AD}(\cdot)$ is not differentiable at 0, the derivative of the Huber norm with some small constant ϵ

$$H'_{\epsilon}(x) = \begin{cases} \frac{x}{\epsilon} & 0 \leq |x| \leq \epsilon \\ \mathrm{sign}(x) & \epsilon < |x| \end{cases} \tag{4.7}$$

is used for the numerical optimization. The integration operation of equation (4.6) does not

depend on the optimization variable $\mathbf{p}$. Thus, denoting

$$h_i := H'_\epsilon\big(I_F^i(\mathbf{x}) - \mathrm{T}^i(I_E(x+u(\mathbf{x}),y),\mathbf{p})\big)\,, \tag{4.8}$$

the gradient of the energy functional with respect to $\mathbf{p}$ reads as

$$\frac{\partial E}{\partial \mathbf{p}} = -\omega \sum_{i=c_1}^{c_m} \int_\Omega h_i \left(\tfrac{\partial \mathrm{T}^i}{\partial p_1}, \quad \ldots \quad, \tfrac{\partial \mathrm{T}^i}{\partial p_k}\right)^T d\mathbf{x}. \tag{4.9}$$

Accordingly, the parameter update for the resulting gradient descent iteration is given by

$$\mathbf{p}_{n+1} := \mathbf{p}_n - \alpha^n \boldsymbol{\lambda} \frac{\partial E}{\partial \mathbf{p}}, \tag{4.10}$$

where $\boldsymbol{\lambda}$ denotes a diagonal weighting matrix for the individual parameter updates. The iteration dependent step size α^n is determined according to the Armijo rule [Arm66], cf. equations (3.29) and (3.30). The repetition of the parameter update until convergence, i.e. until the decrease of energy from n to $n+1$ drops below a certain threshold, finally leads to the desired solution of the optimization problem (4.6).

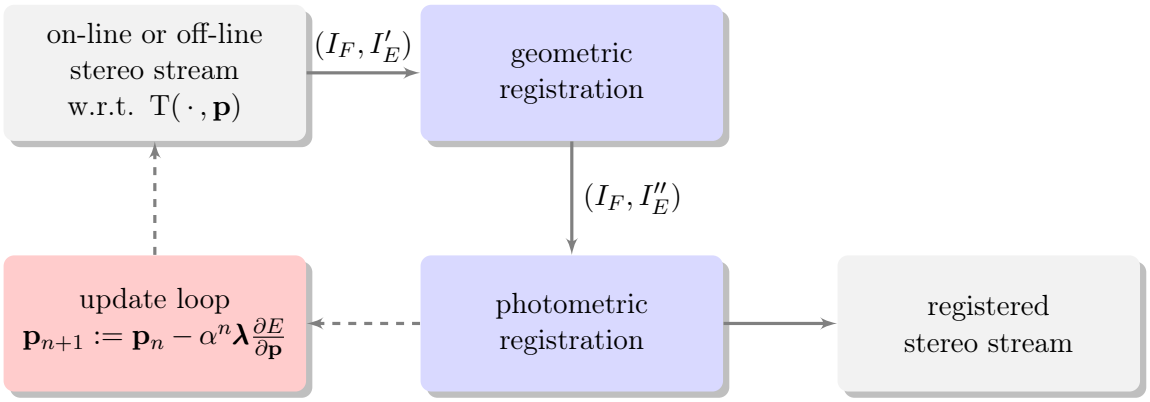


Figure 4.1: Structure of the registration algorithm. The image pair (I_F, I'_E), where $I'_E = T(I_E, \mathbf{p})$, is subject to geometric registration optimization in the first step. In the second step, photometric registration is applied to (I_F, I''_E) with $I''_E(\mathbf{x}) = I'_E(x+u(\mathbf{x}), y)$. A feedback loop takes care about the color adjustment of the input images.

4.4 Parameter Optimization

Considering sections 4.2 and 4.3, it can be expected that the geometric image registration results benefit from a given, precise photometric image registration mapping and vice versa. Therefore, both procedures are merged into an alternating iterative minimization scheme. Each iteration cycle consists of a small fixed number of iterations to solve for u and one iteration to solve for $\mathbf{p}$. Figure 4.1 illustrates the structure of the depth driven registration. The input for the registration process is an on-line or recorded stereo image pair or video stream, where the second image is transformed according to the application specific color distortion function. Based on the input image pair, the geometric registration is performed

as outlined in section 4.2. However, instead of iterating until convergence, only a small fixed number of primal-dual proximal point iterations is conducted in order to allow for a combined convergence of geometric and photometric registration. Thereby, photometric registration biases caused through the interdependency of both registration processes can be avoided. Afterwards, the intermediate primal and dual variables of the proximal point iteration are stored as initial values for the next input stereo pair, and a geometrically registered version of the second image is passed to the photometric registration step. Here, an update of the color transformation parameter vector of the application specific color distortion model is computed and handed over to the input module in order to update the color transformation mapping for the second image. Depending on the actual use case, the parameter update can be applied to an electronic transformation mapping for off-line processing or directly for the adjustment of camera interface parameters in an on-line setup. Subsequently, the entire procedure is repeated with updated color transformation parameters until both registration steps are converged. Formally, the algorithm is considered to reach a converged state if the reduction of energy from one iteration to the next drops below a certain threshold.

Figure 4.2: Two example images of the stereo input for photometric alignment.

4.5 Experiments

In this section, the algorithmic efficiency of the proposed photometric alignment is demonstrated on four different real-world datasets that were recorded with two Ximea CB200CG-CM cameras as listed in table A.1. An overview to these datasets is provided in appendix B. In order to simulate on-line processing, the image data was captured as a completely untouched 12 bit raw readout from the cameras' bayer sensors. By design, the Ximea CB200CG-CM do not allow any camera-based color processing while reading raw data. In consequence, a software-based color processing of the recorded sequences constitutes an on-line equivalent setup.

In the following, section 4.5.1 introduces the applied color transformation and its performance is illustrated qualitatively. Subsequently, its impact on the results of the Patch-Sweep algorithm is evaluated in section 4.5.2.

4.5.1 Affine RGB Registration

Within this work, the goal of the photometric adjustment is twofold. On the one hand, the texture values from different cameras should be photometrically consistent in order to enable for a view synthesis that is based on multiple cameras. On the other hand, the algorithmic complexity of the 3D estimation should be reduced by replacing NCC with a computationally cheaper similarity measure such as the sum of absolute differences (SAD), cf. equation (3.25). Formally, the SAD of two vectors $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^N$ is given by

$$\mathrm{SAD}(\mathbf{X}, \mathbf{Y}) = \frac{1}{N} \sum_{i=1}^{N} |x_i - y_i| \,. \tag{4.11}$$

Within the targeted video communication setup, both goals can be achieved by a registration with a general purpose affine RGB mapping that is used within the presented optimization framework. Formally, this color mapping reads as

$$\mathrm{T}(r, g, b, \mathrm{vec}(\mathbf{A})^T, \mathbf{t}^T) = \mathbf{A} \begin{pmatrix} r \\ g \\ b \end{pmatrix} + \mathbf{t}. \tag{4.12}$$

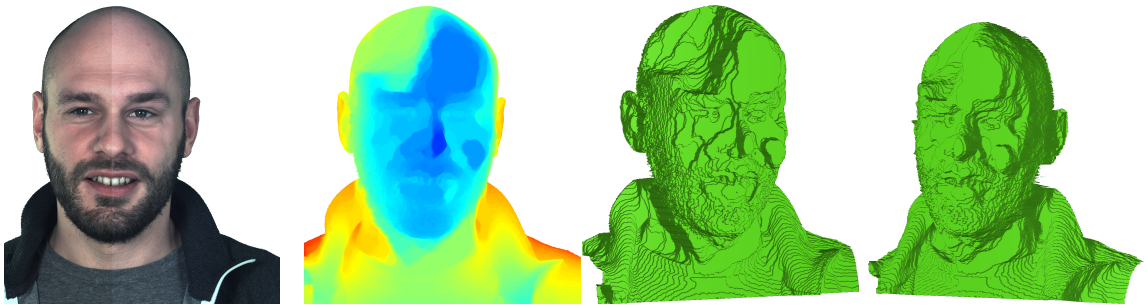
The variable $\mathbf{A}$ denotes an arbitrary 3×3 matrix and $\mathbf{t}$ a translation vector in RGB space. According to equation (4.10), the update of the parameter vector $\mathbf{p} = \left(\mathrm{vec}(\mathbf{A})^T, \mathbf{t}^T\right)^T$ is given by

$$\mathbf{p}_{n+1} \coloneqq \mathbf{p}_n + \alpha^n \boldsymbol{\lambda} \omega \int_{\Omega} \begin{pmatrix} \mathbf{V} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{V} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{V} \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} h_1 \\ h_2 \\ h_3 \end{pmatrix} d\mathbf{x}, \tag{4.13}$$

where $\mathbf{V} = I_E^T(x + u(\mathbf{x}), y)$, $\mathbf{0} = (0, 0, 0)^T$ and the initial parameter vector is the identity mapping $\mathbf{p}_0 = (1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0)^T$. Please note that other more intricated nonlinear color mappings such as gamma correction could also be used if required. Even a lighting model could be applied due to the availability of 3D data.

The qualitative performance is demonstrated on the `SaschaHR` dataset as listed in appendix B. Examples for the input data are provided in figure 4.2. It can be seen that there are colorimetric differences between the two textures and that the highlights on the forehead of the person in the left view are significantly stronger. During the optimization, the right input image is registered to the left input image. Saturated values are clamped during each iteration of the photometric registration. A qualitative illustration of the results is provided in figure 4.3. In the very left column on the top, the difference between the input and the adjusted version of the right image is shown in terms of a split image. It can be seen that there are significant colorimetric changes during registration. At the bottom, there is another split image that shows the left input and a rendered version of the left input that was synthesized based on the color corrected right input. In the second column, there are depth map representations of the results for the geometric registration. The result on top was computed with the identity color mapping and the result at the bottom with the colorimetrically aligned textures. In the third and the fourth column, there are the respective depth-mesh representations. It can be seen that the uncorrected input causes significant errors during registration. Especially the forehead and the cheek of the person are strongly

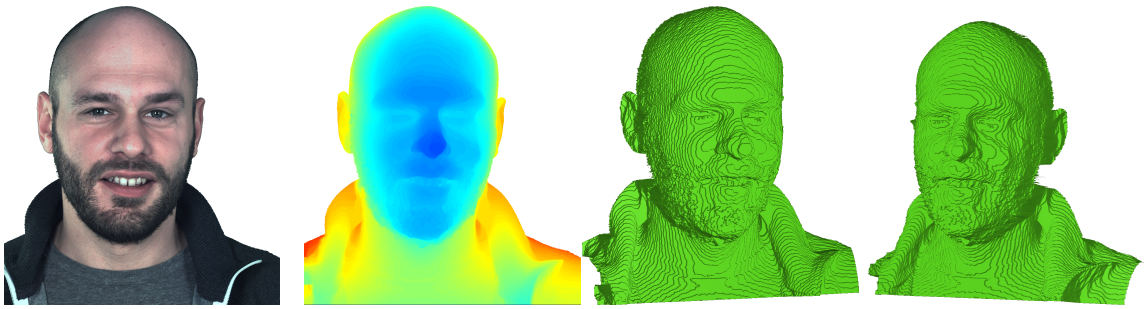
deformed. Other image regions exhibit additional improper but less severe registration errors. In contrast, the corrected results do not suffer from these artifacts and the synthesized view constitutes a very close approximation to the real left input. In particular, it can be seen that the proposed algorithm was able to correct for the negative impact of the facial highlights.



Split: Right input and adjusted right input.

Depth for un-aligned input.

Depth-mesh for un-aligned input.



Split: Left input and synthesized left input based on adjustment.

Depth for aligned input.

Depth-mesh for aligned input.

Figure 4.3: Comparison of textures and depth results that were computed with and without photometric alignment.

4.5.2 Impact on 3D Estimation

As it was demonstrated in section 4.5.1, the proposed photometric calibration significantly improves the results of the globally optimal 3D estimation that was used for the alternating iteration scheme. In this section, the impact of the photometric alignment on the IPS-DEP results is evaluated. For this purpose, three configurations are evaluated on the `SaschaHR`, `RonnyHR`, `HannesHR` and `JohannesHR` datasets: IPS-DEP in combination with NCC and SAD on uncorrected input and IPS-DEP with SAD on corrected input. The selected consistency threshold for all configurations is $T_c = 10\,\text{mm}$. Debayering and white balancing is conducted on graphics hardware upfront to 3D processing. In figure 4.5, examples of the achieved results for the three configurations are shown together with the white balanced and debayered but unaligned stereo input and the right input that was photometrically aligned to the left input. Since NCC is invariant to illumination changes, the results on the very left of the figure serves as reference for comparison with the SAD results. Regarding the uncorrected input, the SAD results are virtually unusable as shown in the center column of the figure. In

contrast, when operating on the aligned stereo pair, SAD results are equivalent to the NCC version as it can be seen on the very right of the figure. The performance of the colorimetric adjustment can also be confirmed numerically. Therefor, depth values and the completeness with respect to the consistency check, as it is described in section 3.2.3.1, are compared. The NCC based results serve as a reference for the SAD computations on uncorrected and corrected input. Only depth values that are considered to be consistent are compared with each other. For a frame wise evaluation, exemplarily the plots of the completeness and the mean absolute depth differences for 500 frames of the `SaschaHR` dataset are provided in figure 4.4. The mean completeness values for all sequences and the mean of the mean absolute depth differences are listed in table 4.1. It can be seen that completeness values for the SAD based results on corrected input images are comparable or superior to those of the NCC results while the mean absolute differences of depth values is only in the range of 2.0 mm. In contrast, the SAD based results for the uncorrected input exhibits a much smaller completeness and much larger mean absolute depth differences. In consequence, it could be verified that the proposed colorimetric alignment algorithm allows for the application of SAD instead of NCC while maintaining the same quality level. Additionally, beside the obvious advantages of aligned textures for texture fusion, the computational load for 3D processing is significantly reduced since SAD requires about three times less arithmetic operations than NCC. The effect of this reduced computational load on computation times of IPS depends mainly on the algorithmic configuration, the image resolution and the graphics card architecture. Depending on the extend of GPU bandwidth limitations for an actual processing task, arithmetic operations are potentially conducted while the GPU has to wait for further input data from video memory. Regarding the processing of the results of table 4.1 with SAD, an average decrease of computation time of about 17 percent compared to NCC have been measured.

	completeness			difference	
	NCC-U	SAD-U	SAD-A	SAD-U	SAD-A
`HannesHR`	90.934	49.306	91.419	88.585	2.630
`RonnyHR`	93.372	64.425	94.485	28.076	1.612
`SaschaHR`	92.107	69.875	94.419	24.926	1.546
`JohannesHR`	89.690	55.390	91.599	72.439	2.326

Table 4.1: Average completeness percentages and average mean depth differences in millimeter. For each listed dataset, 500 frames were evaluated. IPS-DEP results are computed on unaligned (U) and aligned (A) input with NCC and SAD as similarity measure. The absolute depth differences were computed with respect to the NCC-U results.

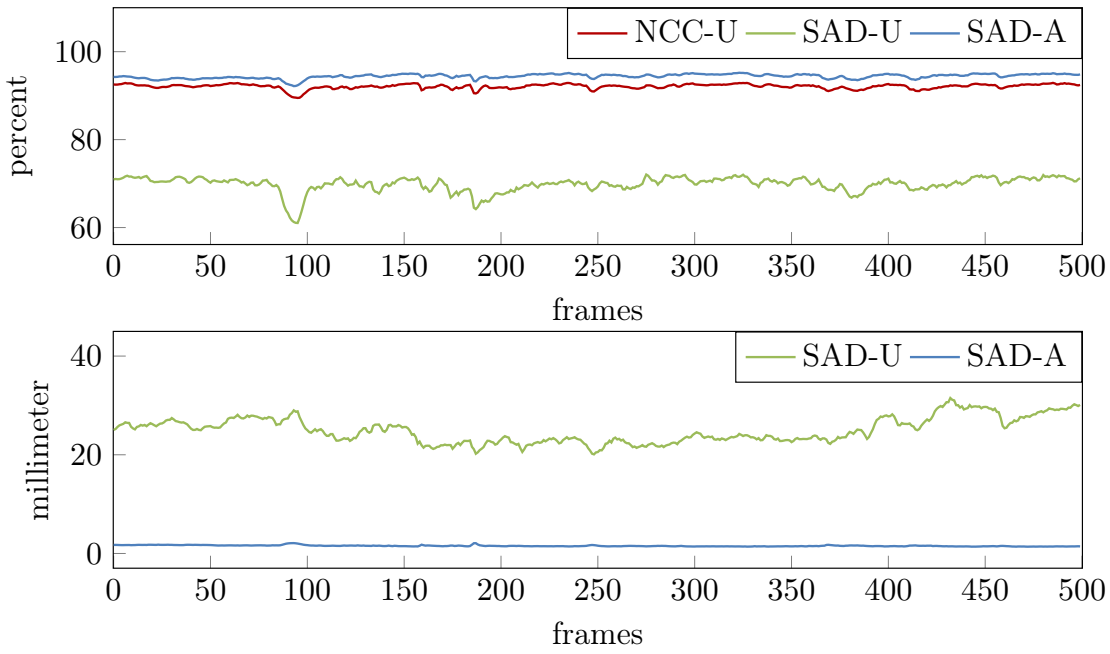


Figure 4.4: Frame wise evaluation for 500 frames of the `SaschaHR` dataset. IPS-DEP results are computed on unaligned (U) and aligned (A) input with NCC and SAD as similarity measure. **Top:** Completeness values with respect to the threshold $T_c = 10\,\text{mm}$. **Bottom:** Frame wise mean absolute differences of depth values that are considered as consistent. The NCC results are used as reference for the depth comparison.

4.6 Chapter Summary

In this chapter, a new algorithm for depth based photometric alignment was presented. The alignment was formulated in terms of an energy minimization problem consisting of a smoothness and a data term. The combined energy term includes both, constraints for the geometric and the colorimetric registration. The generic mathematical formulation allows for the application of an arbitrary color mapping. An alternating optimization was proposed to minimize the energy term. While a primal-dual proximal point iteration algorithm was used to solve for the unknown function that performs the geometric registration, a gradient descent optimization was used to find the optimal parameters for the color mapping. For the evaluation of the algorithmic performance of the proposed approach, in the experiment section an affine RGB transformation was used for color mapping. Based on video communication datasets, the successful photometric alignment was demonstrated. In addition, the impact on 3D estimation with the IPS algorithm was evaluated. It could be shown that aligned input views allow for the application of the computationally less expensive SAD similarity measure instead of the costly NCC.

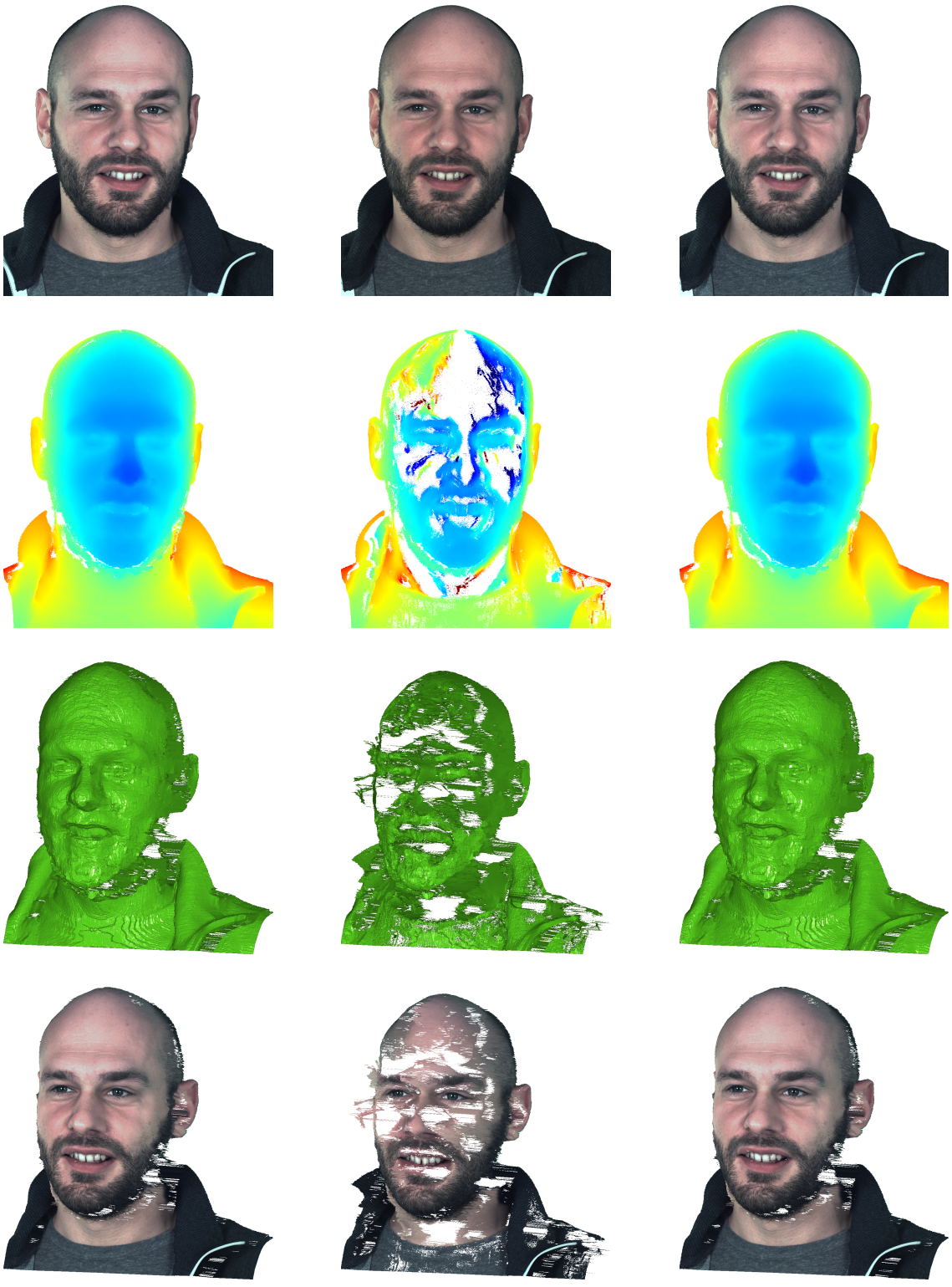


Figure 4.5: Comparison of 3D estimation results for different similarity measures with and without colorimetric registration. **Top row:** From left to right, left texture, uncorrected right texture, corrected right texture. **Second to fourth row:** Result representation in terms of depth maps, polygonal depth-mesh and textured polygonal depth-mesh. Inconsistent results are indicated by white areas. **Results from left to right:** NCC + uncorrected input, SAD + uncorrected input, SAD + corrected input.

5. Eye Contact Provision

In this thesis, a solution to the eye contact problem is presented that is based on real-time 3D reconstruction of the conferees and the subsequent synthesis of virtual eye contact views. However, the provision of convincing eye contact not only depends on the quality of the 3D analysis or the quality of the view rendering, but also on the accuracy of the estimated *eye contact camera*. In order to achieve direct eye contact, the virtual cameras need to be aligned to the line of sight between the two conferees. However, as a prerequisite for identifying the line of sight, the relative position of two conferees need to be known.

The main idea of this chapter consists of the registration of the conferees' coordinate systems and a constitutive continuous update of the current virtual cameras via line of sight estimation. The registration is performed by identifying the rigid 3D transformation that maps both conferees into a common coordinate system. In this way the frame of reference for the line of sight computation is available. The key components for the estimation of this mapping are the available information about display and camera geometry and the eye positions of the conferees that are computed via a triangulation of eye tracking results in multiple views. As the display and the cameras are fixed components, the coordinate system transformation needs to be only recomputed in case a new user is present. This recomputation is a short procedure of only a few seconds prior to video communication. As a manual interaction or adjustment is difficult for unexperienced users, the proposed approach is designed to automatically adapt to arbitrary new users. In addition, the user specific computation of the coordinate mapping provides flexibility for the conferees' sitting positions and avoids predefined and potentially inappropriate *sweet spots*. Once both persons are located within a common coordinate system, the line of sight can be identified as the 3D line that connects the eyes of the conferees and the *eye contact cameras* can be placed accordingly. At this point, an initial eye contact situation with custom *sweep spots* is established. Here, it would be possible to fix the position of the *eye contact cameras* during the conversation. However, this results in the *Mona Lisa effect*. Independent of the viewing perspective of the local conferee, the remote conferee is always rendered from the eye contact perspective if he or she is located at his or her respective *sweep spot*. Hence, in this chapter, it is proposed to conserve more naturalness of the conversation by a continuous update of the line of sight between the conferees and a constitutive update of the individual *eye contact cameras*. In consequence, a conferee is able to circumnavigate his or her chat partner to the extent that is supported by the 3D data that underlies the view rendering.

This chapter is organized in two parts. First, section 5.1 explains the general concept for eye contact provision and outlines the geometric situation and the involved constraints and prerequisites for *eye contact camera* calibration from a theoretical perspective. In particular, this section serves as a conceptual reference for the subsequent algorithmic discussion. And second, in section 5.2, the algorithmic details for the calibration of the *eye contact cameras* are provided. Parts of this chapter have already been published in [Wai+12, 13c].

5.1 Eye Contact Geometry

Regarding provision of eye contact, there are three terms that need to be discriminated prior to the algorithmic discussion. First, *gaze direction* indicates the look at direction of a person. Second, *head pose* denotes the orientation of a persons head. And third, *conversation direction* denotes the line of sight between two persons that are talking to each other. All of these three directions could possibly coincide, but in general they do not have to. During a conversation, persons might deliberately move their eyes or head in order to interrupt eye contact while talking. For the purpose of maintaining natural conversation, in this context, the provision of eye contact is particularly connected to the estimation of the conversation direction. While the gaze and the head pose are pure geometric quantities that can be estimated via image processing algorithms, the identification of the conversation direction needs prior knowledge. If more than two persons are involved, for the estimation of the conversation direction, a semantic interaction analysis would be required. However, focusing on point-to-point video communication, this restriction is not severe, and it can be assumed that the conversation direction is always equal to the line of sight between the two conferees. Consequently, correct eye contact will be generated if and only if the virtual cameras on both sides are aligned with the line of sight. In order to conserve as much naturalness as possible, the two conferees need to be placed into a mutual coordinate frame that allows for a meaningful line of sight computation. This coordinate frame allows for a virtual environment that appropriately represents and preserves the respective real-life face-to-face situation and geometry. In this context, the different aspects of display and conferee position and orientation are discussed separately step-by-step. In section 5.1.1, the relative orientation of the display and the conferee is discussed and the distortions that might occur as a function of the viewing perspective are explained and formulated mathematically. Afterwards in section 5.1.2, the different possible relative positions of both conferees and their displays are discussed and expressed mathematically.

5.1.1 Display Orientation

On top of figure 5.1, a real-life eye contact situation is depicted. Both persons share exactly the same eye height. On the bottom of figure 5.1, on each side a person is replaced by a rendered virtual view. But even if it is assumed that the synthetic view is correctly rendered with respect to the virtual cameras **A** and **B**, the conferees only have a correct and undistorted viewing experience if the displays are mounted in parallel to the image planes of the virtual cameras. As illustrated in figure 5.2, the viewing rays for the real-life

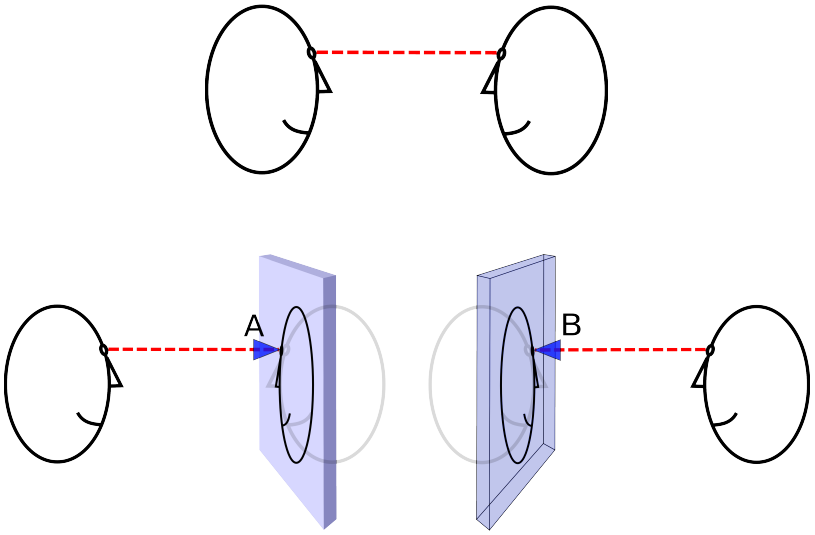


Figure 5.1: Top) Ideal situation regarding virtual camera placement. Both conferees share the same eye height and perceive direct eye contact. **Bottom)** Corresponding remote situation were the respective chat partners are rendered on displays. The eyes of the participants are replaced by the virtual *eye contact cameras* **A** an **B**. The remote conferee is rendered for both sides on equal eye height exactly like in the real-life face-to-face situation.

situation and the virtual setup do not correspond in case of slanted display mounting. The space point $\mathbf{O}$ on the surface of the real conferee is imaged by the *eye contact camera* $\mathbf{B}$ onto the pixel coordinate $\mathbf{x}$. Equally, the corresponding point $\mathbf{X}$ rendered on the parallel display plane $\boldsymbol{\pi}$ is projected onto $\mathbf{x}$ by the conferees eye. But if the image of the *eye contact camera* is rendered onto the display plane $\boldsymbol{\pi}'$, the local conferee might experience an unrealistic perspective, since the rendered point $\mathbf{X}$ on $\boldsymbol{\pi}'$ projects onto $\mathbf{w}$. The extend of this particular distortion might be neglected for small deviations from ideal conditions and the impact to human visual comfort is subject to human factors experiments. However, in order to provide a complete survey on the geometric situation the mathematical perspective on these distortions is provided in the following. The central requirement for a valid rendering is an exact match of the image from the *eye contact camera* for a real-world conferee and the image taken from the rendered view on the display as it would be the case for a display mounted in parallel to the image plane of the *eye contact camera*. Referring to figure 5.2, the mathematical constraint reads as

$$\mathbf{P_B O} \sim \mathbf{P_B X}, \tag{5.1}$$

where $\mathbf{P_B}$ denotes the projection matrix of camera $\mathbf{B}$ and $\mathbf{X}$ may reside on an arbitrary display plane. It is directly implied that the perceived image of the rendered image must match the rendered image. Therefore,

$$0 = \boldsymbol{\pi}' \begin{pmatrix} \lambda \mathbf{K}^{-1} \mathbf{x} \\ 1 \end{pmatrix} = \boldsymbol{\pi}' \mathbf{X}' \tag{5.2}$$

must hold for an arbitrary slanted display plane $\boldsymbol{\pi}'$ and some $\lambda \in \mathbb{R}$. Without loss of generality it can be assumed that

$$\begin{aligned} \mathbf{P_B} &= (\mathbf{K}|\,\mathbf{0})\,, \\ \mathbf{v} &= \begin{pmatrix} 0 & 0 & z \end{pmatrix}^T \text{ and} \\ \boldsymbol{\pi} &= \begin{pmatrix} 0 & 0 & 1 & -z \end{pmatrix}. \end{aligned} \tag{5.3}$$

The rotation axis $\mathbf{a}$ and the angle α between $\boldsymbol{\pi}$ and $\boldsymbol{\pi}'$ allow the computation of the rotation matrix $\mathbf{R}(\mathbf{a}, \alpha) = \mathbf{R}$. For an arbitrary point $\mathbf{U}$ on $\boldsymbol{\pi}$, i.e. $\boldsymbol{\pi}\mathbf{U} = 0$, the transformation from $\boldsymbol{\pi}$ to $\boldsymbol{\pi}'$ reads as

$$\underbrace{\boldsymbol{\pi}\mathbf{H}^{-1}}_{\boldsymbol{\pi}'}\underbrace{\mathbf{H}\mathbf{U}}_{\mathbf{U}'} = 0, \text{ with } \mathbf{H} = \begin{pmatrix} \mathbf{R} & -\mathbf{R}\left(\mathbf{v} - \mathbf{R}^T\mathbf{v}\right) \\ \mathbf{0}^T & 1 \end{pmatrix}, \text{ and } \mathbf{H}^{-1} = \begin{pmatrix} \mathbf{R}^T & \mathbf{v} - \mathbf{R}^T\mathbf{v} \\ \mathbf{0}^T & 1 \end{pmatrix}.$$

Substituting $\boldsymbol{\pi}'$ in equation (5.2) leads to

$$\begin{aligned} 0 &= \boldsymbol{\pi}\mathbf{H}^{-1} \begin{pmatrix} \mathbf{K}^{-1} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} \begin{pmatrix} \lambda\mathbf{x} \\ 1 \end{pmatrix} \\ &= \lambda\mathbf{e}_3^T\mathbf{R}^T\mathbf{K}^{-1}\mathbf{x} + \mathbf{e}_3^T\left(\mathbf{v} - \mathbf{R}^T\mathbf{v}\right) - z \\ \Leftrightarrow \lambda &= \lambda(\mathbf{x}) = z\mathbf{e}_3^T\mathbf{R}^T\mathbf{e}_3 \,/ \mathbf{e}_3^T\mathbf{R}^T\mathbf{K}^{-1}\mathbf{x}\,. \end{aligned} \tag{5.4}$$

Based on the distortion model, a correction can be accomplished by moving the *eye contact camera* according to $\boldsymbol{\pi}'$, i.e. $\mathbf{P}'_\mathbf{B} = \mathbf{P_B}\mathbf{H}^{-1}$, and projecting $\mathbf{X}'$ onto its image plane. In consequence, a corrected image coordinate $\mathbf{x}'$ can be computed as

$$\begin{aligned} \mathbf{x}' &= \mathbf{P}'_\mathbf{B} \begin{pmatrix} \lambda\mathbf{K}^{-1}\mathbf{x} \\ 1 \end{pmatrix} \\ &= \lambda\mathbf{K}\mathbf{R}^T\mathbf{K}^{-1}\mathbf{x} + \mathbf{K}\left(\mathbf{v} - \mathbf{R}^T\mathbf{v}\right). \end{aligned} \tag{5.5}$$

5.1.2 Conferee Position

Beside the *ideal* situation depicted in figure 5.1, real-life persons differ in their eye height. Additionally, a horizontal deviation should be taken into account according to figure 5.3. Regarding real-life conversations, these parameters change continuously. In the following, the influence of these values on the orientation and position, i.e. on the external parameters of the *eye contact camera*, will be discussed as well as the impact of the finiteness of display planes and non equal distances between the conferees and their individual displays as depicted in figure 5.4.

First, unlimited display planes and exactly equal viewing distances for both participants are assumed. For this case, referring to figure 5.3, the external parameters for both *eye contact cameras* $\mathbf{A}$ and $\mathbf{B}$ can be directly derived. The camera centers need to be the centers of the conferees eyes $\mathbf{c_A}$ and $\mathbf{c_B}$ respectively and the orientations are $\mathbf{R_A} := \mathbf{R_A}(\mathbf{a}, \mathbf{v})$ and

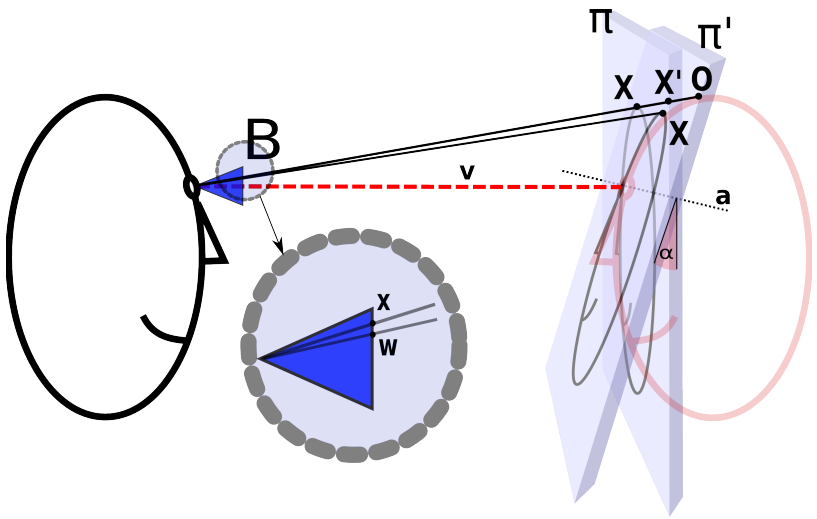


Figure 5.2: The viewing rays of the original 3D object and the rendered view match if the display plane is parallel to the image plane of the *eye contact camera*. Slanted display planes cause a distorted viewing experience.

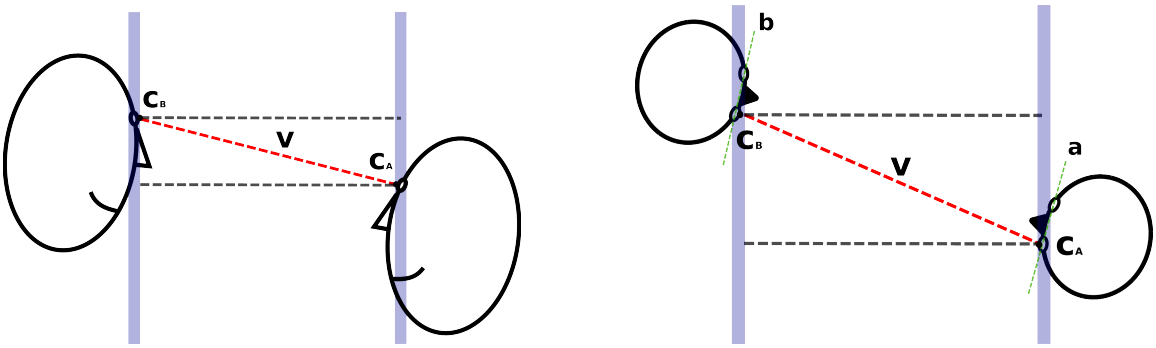


Figure 5.3: Deviations from the situation as depicted in figure 5.1 occur frequently in real-life face-to-face communication. **Left)** Vertical deviation caused by unequal eye height. **Right)** Horizontal deviation, e.g. due to head rotation or body movement.

$\mathbf{R_B} := \mathbf{R_B}(\mathbf{b}, -\mathbf{v})$, where the rotation matrices are determined by the directional vector $\mathbf{v}$ and the *roll vectors* $\mathbf{a}$, $\mathbf{b}$, i.e. the projection of each vector must be parallel to the x-axis on the corresponding image plane. However, taking non equal viewing distances into account as illustrated in figure 5.4, a different approach is needed since inconsistent viewing directions $\mathbf{v_A}$, $\mathbf{v_B}$ emerge. The proposed solution to this problem is depicted in figure 5.5. Instead of placing the displays on the positions of the remote conferees, both screens are equally positioned directly on the viewing ray together with the *eye contact cameras* according to the participants viewing directions. Thus, different distances are supported while a single natural eye contact direction is preserved. The external camera parameters can be directly assembled and read as $\mathbf{c_{A/B}} = \mathbf{c_A} = \mathbf{c_B}$ and $\mathbf{R_A} := \mathbf{R_A}(\mathbf{a}, \mathbf{v_A})$, $\mathbf{R_B} := \mathbf{R_B}(\mathbf{b}, \mathbf{v_B})$. But the internal camera parameters need to be adjusted according to the respective distances. Let f be the focal length of the *eye contact cameras* for the setup in figure 5.3. By placing the displays somewhere on the eye contact viewing ray, the proportional rendering size has

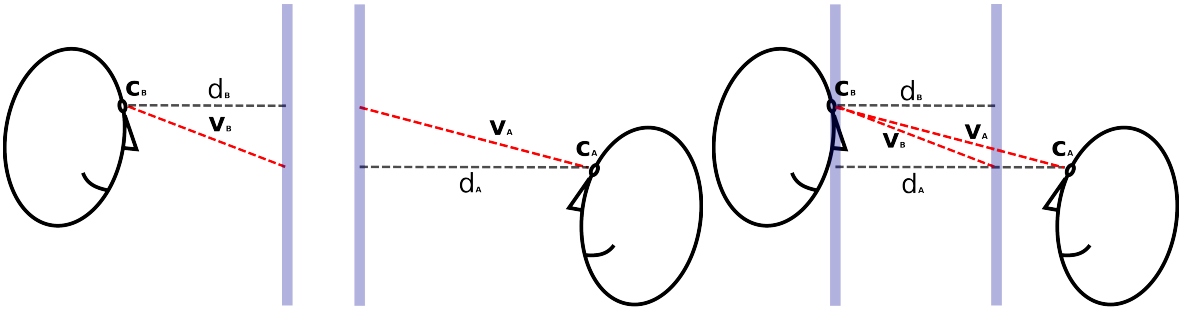


Figure 5.4: Different viewing distances for each remote conferee need to be considered. **Left)** Two conferees with different viewing distances. **Right)** Overlay of left illustrations shows a line of sight conflict.

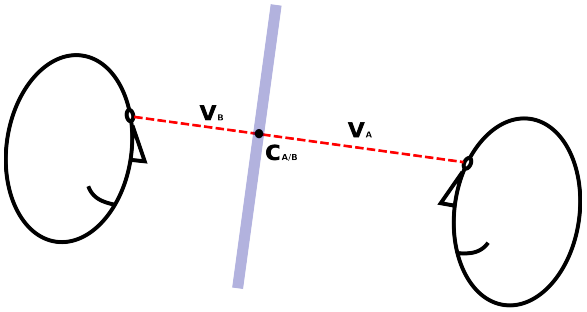


Figure 5.5: The concept of proportional rendering size maintenance naturally supports different viewing distances.

to be maintained, i.e.

$$\begin{aligned} \frac{f_{\mathbf{A}}}{f_{\mathbf{B}}} &= \frac{\|\mathbf{v_A}\|}{\|\mathbf{v_B}\|} \\ \Rightarrow f_{\mathbf{A}} &= f\frac{\|\mathbf{v_A}\|}{\|\mathbf{v_B}\|} \\ \Rightarrow f_{\mathbf{B}} &= f\left(2 - \frac{\|\mathbf{v_A}\|}{\|\mathbf{v_B}\|}\right), \end{aligned} \tag{5.6}$$

where $f_{\mathbf{A}}$ and $f_{\mathbf{B}}$ are the focal lengths of the *eye contact cameras* **A** and **B**. Finally, depending on the applied displays, limitations can occur in case that the real screens do not share a common virtual display area with the virtual screen placed on the viewing ray as illustrated in figure 5.6. While such a situation is very unlikely for display wall setups, it might happen for desktop-sized screens in combination with extreme positioning schemes. However, if a proper screen setup is not possible, the *eye contact cameras* cannot act like the real eyes of the conferees, but eye contact can be still achieved to some extent by accepting the different viewing directions $\mathbf{v_A}$, $\mathbf{v_B}$ and camera centers $\mathbf{c_A} \neq \mathbf{c_B}$ for the computation of the externals. But the result will comprise an inherent conflict. E.g. in figure 5.6, both conferees *look down* to their remote side, but they have to be recorded from a *look up* perspective in order to catch the eye contact view. This leads to a unnatural conversation experience where both conferees *look down* to someone who is rendered like *looking up* while the *looking down* bearings are maintained.

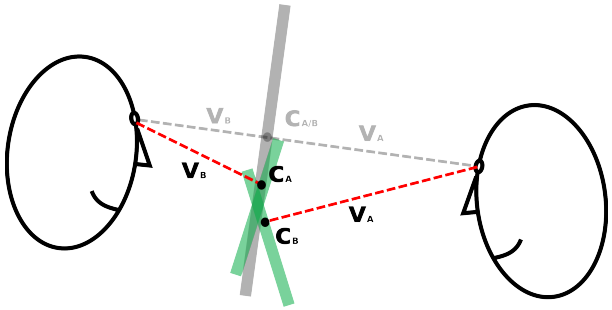


Figure 5.6: Inappropriate display positions especially in combination with small displays may easily lead to geometrical contradictions. Eye contact is possible to some extent, but the conferees bearings conflict with the virtual view rendering.

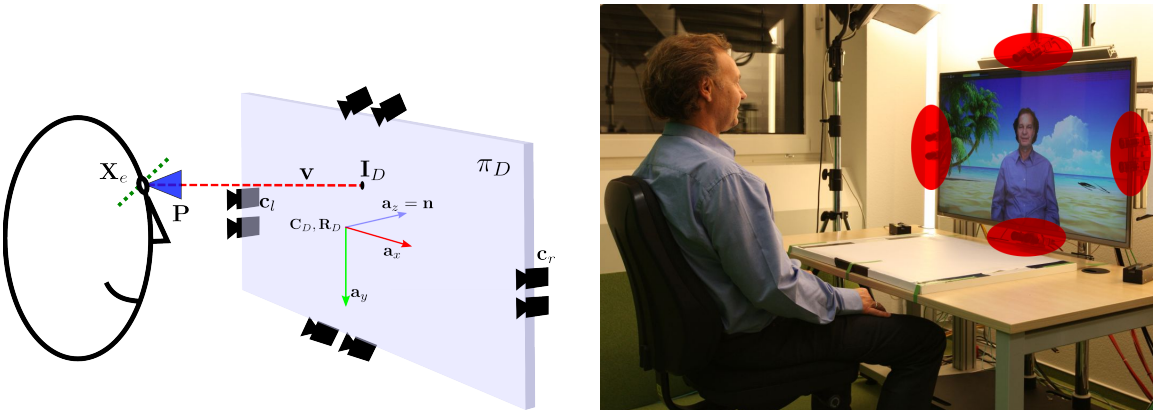


Figure 5.7: Left: Schematic illustration of the display plane together with the relevant geometric entities. **Right:** Camera setup used for experiments. The cameras' mounting positions are highlighted in red. Cameras mounted at the display frame allow for the computation of the display plane via the camera centers.

5.2 The Eye Contact Camera

In section 5.1, eye contact provision has been addressed from a theoretical point of view. Constitutively, in the following, the computation of the eye contact cameras for point-to-point video communication in a real-world scenario is discussed. The camera estimation is motivated based on the setup that is illustrated in figure 5.7. However, apart from the setup specific computation of the display coordinate system, the presented eye contact camera estimation is independent from display and camera configurations. According to the structural diagram in figure 5.8, the process of eye contact camera estimation can be divided into three parts.

First, when setting up the video communication system, on both remote sides the local *display geometry* needs to be identified once in order to enable for a correct positioning of the rendered conferees. The metric display coordinate system is computed constitutively. It encodes the relative position and orientation of the display and the cameras. In order to enable for the conversion from metric to pixel coordinates and vice versa, the metric width and height ($D_{mw} \times D_{mh}$) and the display resolution in pixel ($D_{pw} \times D_{ph}$) need to be looked up from the manual of the installed display type. These values are referred to as display

properties. The computation of the *display geometry* is addressed in section 5.2.1.

Second, based on the display coordinate system, the display properties and the individual line of sight from the respective remote side, a coordinate system mapping between the remote sides is computed. The resulting joint coordinate system represents the common virtual communication environment. The algorithmic computation of the coordinate alignment is discussed in section 5.2.2.

Third, based on the continuous update of the 3D eye positions from both remote sides, the line of sight within the virtual communication environment is updated in order to enable for a match between the viewing and the rendering perspective. The respective assembly of the eye contact camera is described in section 5.2.3.

The 3D eye positions required for coordinate alignment and line of sight update are continuously computed via a triangulation of the results of a state of the art eye tracker [KE06] that is applied to multiple views. A subsequent bundle adjustment is carried out in order to minimize the re-projection error of the triangulation results [HZ04, LA09].

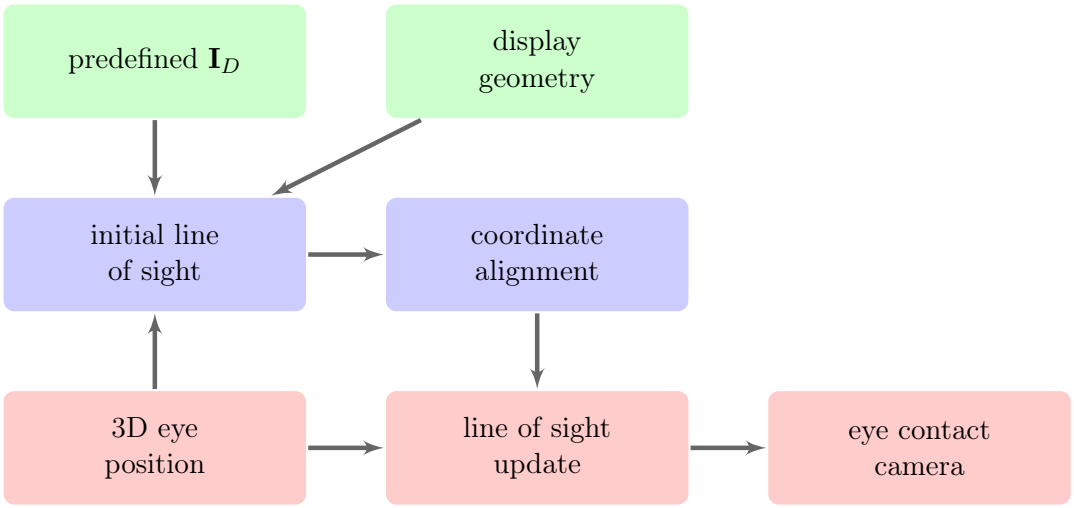


Figure 5.8: Structure of the components for eye contact provision. Green indicates fixed entities, blue is used for entities that need to be estimated only once and red indicates the need or possibility of a continuous update.

5.2.1 Display Geometry

Preliminary to the algorithmic discussion, the required metric geometric entities are introduced as illustrated in figure 5.7. On the display plane $\boldsymbol{\pi}_D$, the display coordinate system is depicted. It consists of $\mathbf{C}_D$ denoting the origin and a rotation matrix $\mathbf{R}_D$ that encodes the orientation of the coordinate axes. The coordinates of the eyes are introduced as $\mathbf{X}_{e_l}$ and $\mathbf{X}_{e_r}$ for the left and the right eye respectively, or $\mathbf{X}_e$, if the distinction between the eyes is not relevant. A priori, the relative position of the display with respect to the real cameras is unknown. However, as outlined in section 5.1, the spatial relationship between conferee, cameras, and display needs to be identified in order to provide a correct rendering of the eye contact view. Many different approaches reaching from manual interaction using measuring equipment to solution with precalibrated cameras that are integrated into the display frame

could be applied to retrieve the required entities $\mathbf{C}_D$, $\mathbf{R}_D$ and $\boldsymbol{\pi}_D$. In this work, a display calibration procedure is proposed that is based on the applied camera arrangement. While this approach can be modified to serve various camera configurations, in the following, the experimental setup illustrated in figure 5.7 is exemplarily discussed. Here, the display plane can be identified by the assumption that all camera centers are located on this plane. The resulting equation

$$0 = \boldsymbol{\pi}_D \left(\mathbf{c}_0, \ldots, \mathbf{c}_{N-1} \right) \tag{5.7}$$

is solved by means of least squares solution approach, where $\mathbf{c}_0, \ldots, \mathbf{c}_{N-1}$ denote the respective camera centers in homogeneous notation. The display plane representation is selected as $\boldsymbol{\pi}_D = \left(\mathbf{n}^T, d \right)$, with unit length normal $\|\mathbf{n}\| = 1$. In order to ensure a unique orientation for the constitutive coordinate system alignment between the remote sides, the direction of the normal of the display plane is set with respect to the computed 3D eye coordinates. As illustrated in figure 5.7, it is expected that the plane normal points away from the eye coordinate $\mathbf{X}_e$ as

$$\boldsymbol{\pi}_D \leftarrow \boldsymbol{\pi}_D \cdot \operatorname{sign} \left(-\frac{\langle \mathbf{X}_e, \mathbf{n} \rangle + d}{\langle \mathbf{n}, \mathbf{n} \rangle} \right). \tag{5.8}$$

Similarly, the regular distribution of the cameras allows for the determination of the metric display coordinate system. The origin $\mathbf{C}_D$ reads as the projection of the center of mass of the camera centers onto the display plane according to

$$\mathbf{C}_D = \operatorname{proj}_{\boldsymbol{\pi}_D} \left(\frac{1}{N} \sum_{i=0}^{N-1} \mathbf{c}_i \right), \tag{5.9}$$

where $\operatorname{proj}_{\boldsymbol{\pi}}(\mathbf{x})$ denotes the orthogonal projection of $\mathbf{x}$ onto the plane $\boldsymbol{\pi}$. The coordinate axes of the display plane are encoded in the rotation matrix $\mathbf{R}_D = (\mathbf{a}_x, \mathbf{a}_y, \mathbf{a}_z)$. As the display plane normalization in equation (5.8) renders the display normal vector into a unique quantity, it can be directly assigned to the $z-$axis of the coordinate system. Further on, the $x-$axis is derived based on two camera centers $\mathbf{c}_l$ and $\mathbf{c}_r$ with appropriate horizontal positions on the left and right display frame as illustrated in figure 5.7. Both camera centers are projected onto the display plane and the normalized directional vector gained by subtracting the projection of $\mathbf{c}_l$ from the projection of $\mathbf{c}_r$ is assigned to $\mathbf{a}_x$. As illustrated in figure 5.7, the final coordinate system axes read as

$$\begin{aligned} \mathbf{a}_x &= \frac{\operatorname{proj}_{\boldsymbol{\pi}_D} (\mathbf{c}_r) - \operatorname{proj}_{\boldsymbol{\pi}_D} (\mathbf{c}_l)}{\| \operatorname{proj}_{\boldsymbol{\pi}_D} (\mathbf{c}_r) - \operatorname{proj}_{\boldsymbol{\pi}_D} (\mathbf{c}_l) \|}, \\ \mathbf{a}_y &= \frac{\mathbf{a}_z \times \mathbf{a}_x}{\| \mathbf{a}_z \times \mathbf{a}_x \|} \text{ and} \\ \mathbf{a}_z &= \mathbf{n}. \end{aligned} \tag{5.10}$$

5.2.2 Virtual Communication Environment

In this section, the alignment of both communication sides to a common 3D coordinate system is presented. This alignment has to be conducted only once prior to video communication. In general, both remote sides are initially unaligned. On each side, there is a partial

line of sight $\mathbf{v_{A/B}}$ from the conferee to the display plane, c.f figure 5.7. As illustrated in figure 5.9, the basic idea of the presented procedure is to register the spatial points where the partial lines of sight intersect the display planes, rotate in order to place the displays in back-to-back position, and straighten the joint line of sight afterwards. The straightened joint line of sight is referred to as *initial line of sight*. Please note that this approach implicitly handles the case of inappropriate display positions as shown in figure 5.6.

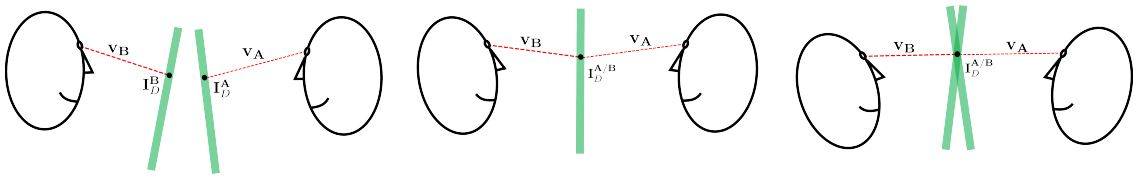


Figure 5.9: Illustration of the mapping into a common coordinate system. **Left)** Initial situation: Two conferees reside in their individual coordinate systems. **Middle)** Back-to-back mapping of display planes. **Right)** The joint line of sight is straightened.

The structural overview in figure 5.8 situates the alignment procedure within the calibration process of the eye contact camera. There are four entities that are directly or indirectly used for the computation of the common coordinate frame. The display geometry as discussed in section 5.2.1, the 3D coordinates of the conferees eyes $\mathbf{X}_e$, predefined intersection points $\mathbf{I}_D$ of the line of sight and the display planes, and the initial line of sight. However, in order to enable for a line of sight within a joint coordinate system, first the partial lines of sight are required. For this purpose, an arbitrary point on the display plane is predefined in terms of pixel coordinates, e.g. the display center. This point is denoted as $\mathbf{I}_D^p$. While a predefinition slightly restricts the user's freedom to choose an initial viewing direction, a major advantage of this approach is the possibility to fully control where the conferee is initially placed on the screen. Therefore, the risk for the rendered remote conferee to overstep the display border while moving is mitigated. Based on the pixel coordinates $\mathbf{I}_D^p$ of the intersection points, the partial lines of sight are computed as the lines that connect the 3D eye coordinates $\mathbf{X}_e$ and the metric 3D coordinates $\mathbf{I}_D$ of the intersection points $\mathbf{I}_D^p$. Based on the display coordinate system and the display properties, $\mathbf{I}_D$ can be computed as

$$\mathbf{I}_D = \mathbf{C}_D + \left(\mathbf{I}_D^p - \begin{pmatrix} \frac{D_{pw}}{2} \\ \frac{D_{ph}}{2} \end{pmatrix} \right) \circ \begin{pmatrix} \frac{D_{mw}}{D_{pw}} \\ \frac{D_{mh}}{D_{ph}} \end{pmatrix}. \tag{5.11}$$

The computation of the Euclidean 3D alignment transformation is based on two requirements. First, both intersection points must coincide, i.e. $\mathbf{I}_D^{\mathbf{A}} = \mathbf{I}_D^{\mathbf{B}}$, and the partial lines of sight from both sides must be collinear. The remaining rotational degree of freedom is eliminated by demanding a unique rotation mapping $\mathbf{R}_D^{\mathbf{AB}}$ between the display coordinate system axes according to

$$\begin{aligned} \mathbf{R}_D^{\mathbf{AB}} \left(\mathbf{a}_x^{\mathbf{B}}, \mathbf{a}_y^{\mathbf{B}}, \mathbf{a}_z^{\mathbf{B}} \right) &= \left(-\mathbf{a}_x^{\mathbf{A}}, +\mathbf{a}_y^{\mathbf{A}}, -\mathbf{a}_z^{\mathbf{A}} \right) \\ \Leftrightarrow \mathbf{R}_D^{\mathbf{AB}} &= \left(-\mathbf{a}_x^{\mathbf{A}}, +\mathbf{a}_y^{\mathbf{A}}, -\mathbf{a}_z^{\mathbf{A}} \right) \left(\mathbf{R}_D^{\mathbf{B}} \right)^T. \end{aligned} \tag{5.12}$$

The line of sight straightening mapping $\mathbf{R_L}$ is computed based on the rotational difference

after normalizing to $\mathbf{I}_D^{\mathbf{A}} = \mathbf{I}_D^{\mathbf{B}} = \mathbf{0}$ and applying $\mathbf{R}_D^{\mathbf{AB}}$. Consequently, the transformation requirement reads as

$$\mathbf{R_L V} = -\frac{\|\mathbf{V}\|}{\|\mathbf{U}\|}\mathbf{U}, \text{ with } \mathbf{V} = \mathbf{R}_D^{\mathbf{AB}}\left(\mathbf{X}_e^{\mathbf{B}} - \mathbf{I}_D^{\mathbf{B}}\right) \text{ and } \mathbf{U} = \mathbf{X}_e^{\mathbf{A}} - \mathbf{I}_D^{\mathbf{A}}. \tag{5.13}$$

The solution to this vector-to-vector mapping can be directly obtained by applying the Rodrigues' rotation formula. The combination of the equations (5.12) and (5.13) leads to the final coordinate system mapping from side **B** to side **A**

$$\mathbf{X}' = \mathbf{R_L R}_D^{\mathbf{AB}}\left(\mathbf{X} - \mathbf{I}_D^{\mathbf{B}}\right) + \mathbf{I}_D^{\mathbf{A}} = \mathbf{T}_D^{\mathbf{AB}}\left(\mathbf{X}\right). \tag{5.14}$$

Similar to the situation in figure 5.5, the Euclidean transformation implicitly defines a virtual *conversation plane* that is orthogonal to the initial line of sight and contains both display intersection points. Please note that aside from a configuration as shown in figure 5.1, in general, the display planes do not superimpose the *conversation plane*.

5.2.3 Line of Sight Update and the Eye Contact Camera

Once the coordinate system mapping is available, i.e. both sides are calibrated, the current eye contact cameras can be computed based on a continuous line of sight update as outlined in figure 5.8. Exemplarily, in the following, the computation of $\mathbf{P_A}$ is discussed. $\mathbf{P_B}$ can be determined analogously. First, according to equation (5.14), the camera center is placed into the transformed remote eye coordinates as

$$\mathbf{c_A} = \mathbf{T}_D^{\mathbf{AB}}\left(\mathbf{X}_{e_l}^{\mathbf{B}}\right). \tag{5.15}$$

The eye contact camera needs to be vectored according to the line of sight direction. In order to avoid rotational ambiguities, the $x-$axis of the camera coordinate system is directed from the left to the right eye. In consequence, with $\mathbf{u} = \mathbf{T}_D^{\mathbf{AB}}\left(\mathbf{X}_{e_r}^{\mathbf{B}}\right) - \mathbf{c_A}$ and $\mathbf{v} = \mathbf{X}_{e_l}^{\mathbf{A}} - \mathbf{c_A}$, the camera rotation matrix reads as

$$\mathbf{R_A} = \begin{pmatrix} \mathbf{r}_0 = & \frac{\mathbf{u} - \langle \mathbf{u}, \mathbf{r}_2 \rangle \mathbf{r}_2}{\|\mathbf{u} - \langle \mathbf{u}, \mathbf{r}_2 \rangle \mathbf{r}_2\|} \\ \mathbf{r}_1 = & \frac{\mathbf{r}_2 \times \mathbf{r}_0}{\|\mathbf{r}_2 \times \mathbf{r}_0\|} \\ \mathbf{r}_2 = & \frac{\mathbf{v}}{\|\mathbf{v}\|}. \end{pmatrix} \tag{5.16}$$

For the rendering of a correct eye contact view, the intrinsic camera parameters need to reflect the geometric relationship between the line of sight and the display position. Especially, the current metric intersection point $\mathbf{I}_D^{\mathbf{A}}$ with the remote display plane needs to be transfered into pixel coordinates in order to determine the principal point of the eye contact camera. Here, $\mathbf{I}_D^{\mathbf{A}}$ is obtained by the intersection of the current line of sight and the remote display plane. The remote coordinates have to be transformed with respect to equation (5.14). Let

$$\begin{aligned} \left(\mathbf{a}_x^{\mathbf{B_T}}, \mathbf{a}_y^{\mathbf{B_T}}, \mathbf{a}_z^{\mathbf{B_T}}\right) &= \mathbf{R_L R}_D^{\mathbf{AB}}\left(\mathbf{a}_x^{\mathbf{B}}, \mathbf{a}_y^{\mathbf{B}}, \mathbf{a}_z^{\mathbf{B}}\right) \\ \mathbf{C}_D^{\mathbf{B_T}} &= \mathbf{T}_D^{\mathbf{AB}}\left(\mathbf{C}_D^{\mathbf{B}}\right) \end{aligned} \tag{5.17}$$

be the transformed remote display coordinate system and

$$\left(\mathbf{n}_{\mathbf{B_T}}^T \; d_{\mathbf{B_T}}\right) = \left(\left(\mathbf{R_L}\mathbf{R}_D^{\mathbf{AB}}\mathbf{n_B}\right)^T \; \mathbf{n}_{\mathbf{B}}^T\left(\mathbf{I}_D^{\mathbf{B}} - \left(\mathbf{R_L}\mathbf{R}_D^{\mathbf{AB}}\right)^T\mathbf{I}_D^{\mathbf{A}}\right) + d_{\mathbf{B}}\right) \tag{5.18}$$

the transformed display plane. The current metric intersection point offset on the display plane reads as

$$\Delta\mathbf{I}_D^{\mathbf{A}} = \mathbf{c_A} - \frac{\langle \mathbf{c_A}, \mathbf{n_{B_T}}\rangle + d_{\mathbf{B_T}}}{\langle \mathbf{r}_2, \mathbf{n_{B_T}}\rangle}\mathbf{r}_2 - \mathbf{C}_D^{\mathbf{B_T}} = \mathbf{I}_D^{\mathbf{A}} - \mathbf{C}_D^{\mathbf{B_T}}. \tag{5.19}$$

Given the metric offset, the pixel offset is obtained by projecting $\Delta\mathbf{I}_D^{\mathbf{A}}$ onto the display coordinate system $x/y-$axes and multiply the result with the pixel per millimeter ratio according to

$$\mathbf{I}_D^{\mathbf{A},p} = \begin{pmatrix} \frac{D_{pw}}{2} \\ \frac{D_{ph}}{2} \end{pmatrix} + \begin{pmatrix} \langle \Delta\mathbf{I}_D^{\mathbf{A}}, \mathbf{a}_x^{\mathbf{B_T}}\rangle \\ \langle \Delta\mathbf{I}_D^{\mathbf{A}}, \mathbf{a}_y^{\mathbf{B_T}}\rangle \end{pmatrix} \circ \begin{pmatrix} \frac{D_{pw}}{D_{mw}} \\ \frac{D_{ph}}{D_{mh}} \end{pmatrix} = \begin{pmatrix} u_x^{\mathbf{A}} \\ u_y^{\mathbf{A}} \end{pmatrix}. \tag{5.20}$$

Finally, complete intrinsics for the eye contact camera can be provided after computing the focal length $f_{\mathbf{A}}$ with respect to the desired rendering proportions, e.g. life size. Based on the scene geometry, i.e. person to display distance, pixel per millimeter ratio and eye distance, the value of the focal length can be directly computed via the intercept theorem. Please note that there is also the possibility of moving the camera along the line of sight, while maintaining the rendering proportions according to equation (5.6). After assembling the parts, the eye contact camera reads as

$$\mathbf{P_A} = \mathbf{K_A}\mathbf{R_A}\left(\mathbf{I}\,|\,-\mathbf{c_A}\right), \text{ with } \mathbf{K_A} = \begin{pmatrix} f_{\mathbf{A}} & 0 & u_x^{\mathbf{A}} \\ 0 & f_{\mathbf{A}} & u_y^{\mathbf{A}} \\ 0 & 0 & 1 \end{pmatrix}. \tag{5.21}$$

Please note that rendering the scene with respect to $\mathbf{P_A}$ leads to adequate results for setups like the one that is illustrated in figure 5.7. However, in case of strongly slanted viewing angles, an additional correction according to equation (5.5) needs to be considered as a part of the rendering process.

5.3 Experiments

In the following, the proposed automatic *eye contact camera* calibration is evaluated on all datasets from the setup that consists of sixteen cameras as listed in appendix B. The goal of the experiments is to illustrate the capabilities of the introduced approach in terms of providing an eye contact view, avoiding a single *sweet spot* and circumnavigating the remote conferee to some extent. In order to simulate a two side conversation situation with off-line data, the calibration process is configured to work in mirror mode, i.e. the calibration is conducted as if the conferees are chatting with themselves. For an illustration of this configuration, please refer to figure A.1. The persons are sitting at a distance of about 600 mm in front of a full HD display with a metric size of 1045 × 600 mm. The initial intersection point $\mathbf{I}_D^p$ of the partial line of sight and display plane is set to the upper

half of the display in centered position, $\mathbf{I}_D^p = (960, 700)^T$. The eye contact views are rendered based on the fused 3D data as presented in section 3.2.4.5. However, in contrast to section 3.2.4.5, where the raw 3D data was fused, a cross bilateral median filter [MZK10, Rie+12b] with a small kernel size of 8×8 pixel is applied with 4 iterations prior to depth fusion. In this way, small holes in the depth data are filled and the results are moderately smoothed. As such filters are computationally cheap and well suited for the implementation on graphics cards, the real-time constraint is not violated. The resulting fused 3D model is rendered in OpenGL with respect to the computed *eye contact camera*. Example eye contact views, together with the closest original views from the cameras that are attached directly at the display frame, are presented in figures 5.10 and 5.12. Regarding the original views, it can be seen that the display size causes a significant displacement from the eye contact perspective, while the virtual view moderates the impression of perceiving direct eye contact. In figure 5.13 the contrast between a continuous *eye contact camera* update and a fixed *eye contact camera* is illustrated. A camera that is predefined or computed once and fixed afterwards leads to a single *sweet spot*. As the person is moving, the eye contact perspective gets lost. The continuous update of the *eye contact camera* allows for an compensation of this effect. Finally, figure 5.11 illustrates the circumnavigation of a remote conferee. While the local conferee is moving to the left and right, the remote conferee is displayed from different perspectives.

5.4 Chapter Summary

This chapter has presented a novel algorithm for the automatic calibration of the eye contact view in a two-sided video communication setup. In contrast to other works, no manual user interaction is required. Prior to the algorithmic presentation, the geometrical constraints for eye contact provision have been discussed in detail. As a prerequisite, the relative position of the display and the cameras needs to be identified once during the setup of the video communication system, e.g. as a part of the camera calibration procedure. Based on this information and the 3D eye positions of the conferees, both communication sides can be mapped into a joint coordinate system. Once the coordinate mapping is computed, the line of sight between the two conferees is continuously updated and the *eye contact cameras* are placed accordingly. Experiments on different video communication datasets were conducted. The presented rendering results for the computed eye contact perspectives exhibit a convincing quality. In addition, the contrast between a fixed *sweet spot* and the proposed continuous *eye contact camera* update has been illustrated, and intermediate views have been shown that were captured during the circumnavigation of a remote conferee.

Figure 5.10: Center: Example eye contact views for the `David`, `Paul` and `Sylvain` datasets. **Right:** Closeups of the eye contact views. **Left:** Examples of original views from cameras that are directly attached to the display frame.

Figure 5.11: Example for a circumnavigation of a remote conferee. The remote conferee keeps his position and conversation direction fixed while the local conferee is moving in order to look at right of the remote conferee. **Left:** Initial eye contact view. **Right:** Three intermediate images during circumnavigation.

Figure 5.12: Center: Example eye contact views for the `Marcus`, `Niklas` and `Oliver2` datasets. **Right:** Closeups of the eye contact views. **Left:** Examples of original views from cameras that are directly attached to the display frame.

Figure 5.13: Contrast between a single *sweet spot* and a continuous update of the *eye contact camera*. From the initial position, the person moved about 300 mm to its left. **Left:** Initial eye contact view. **Middle:** View after the movement, rendered with the fixed initial *eye contact camera*. The person left its *sweet spot*. **Right:** View after the movement, rendered with an updated *eye contact camera*.

6. Summary and Conclusion

In this thesis, a solution for the eye contact problem has been presented that is based on novel algorithms for real-time 3D analysis from multiple views, a depth based photometric image registration, and a continuous update of the eye contact camera. The real-time constraints for the computationally demanding 3D processing have been addressed with an efficient and highly parallelizable algorithmic approach that allows for an implementation on modern graphics cards and many-core systems. As a preprocessing step for eye contact rendering, the computed 3D data is combined in terms of patch groups. The photometric image registration enables for a seamless texturing of the patch group representation with textures from different views. In contrast to other works, the eye contact view is continuously updated based on line of sight information that is computed from 3D eye positions. In consequence, a single *sweet spot* is avoided. Additionally, the conferees are able to virtually circumnavigate their chat partner to the extent that is supported from the available 3D data. All presented algorithms have been evaluated on multiple video communication datasets, and the functional efficiency has been demonstrated. In particular, it has been shown that the rendered eye contact views exhibit an adequate quality. In addition to the off-line evaluation on prerecorded datasets, the proposed algorithms were part of demonstrator setups for public exhibitions and lab demonstrations, c.f. appendix A.3.

Two algorithms for 3D processing have been proposed, a new variant of an Image Based Visual Hull and a novel patch-based stereo algorithm that has been denoted as Patch-Sweep. The algorithmic developments are focused on efficiency and parallelizability in order to exploit the massive computational power of modern many-core systems and graphics cards. While both algorithms could be applied individually, a potential combination has been proposed that leads to a significantly improved completeness of the 3D estimation results, especially in case of less textured image regions. Beside parallel processing, the functional efficiency of the IBVH algorithm is based on a novel cache structure for line segment intersection tests and a constitutive pixel preselection in the desired image. In addition, object cutoffs at the image border are addressed with an interval extension strategy that prevents cutoffs in the final 3D result. The computational speed and the quality of results have been evaluated with video communication datasets. It has been shown that high resolution input from 16 cameras can be processed within real-time on a single graphics card. The Patch-Sweep algorithm is based on the evaluation of hypotheses lists in terms of 3D patches. Different patch representations have been proposed. As a reference for the evaluation of results, an exhaustive Patch-Sweep variant has been introduced. Subsequently,

the computationally more efficient Iterative Patch-Sweep has been presented. It is a 3D estimation framework that consists of three exchangeable major building blocks for the representation, the update, and the propagation of hypotheses. Results from previous frames are used to initialize the current iteration, and the hypotheses are propagated within a predefined neighborhood in image space. New hypotheses are generated by updating previous results with a Monte Carlo approach. Different options for the selection of the three major components have been presented. Based on real-world video communication datasets, these components have been evaluated with respect to the resulting rate of convergence and the quality of results. The efficiency of the proposed Monte Carlo hypotheses update has been compared with a deterministic, numeric steepest gradient approach. In addition, the impact of different neighborhoods for the propagation of hypotheses and the impact of multi-scale processing on the rate of convergence have been investigated. Among the vast amount of possible parameter settings, an optimal algorithmic configuration was empirically identified within a reasonable parameter range. Here, the completeness of the results with respect to a certain quality threshold served as a benchmark for parameter selection. In conjunction with multi-scale processing, a single iteration is commonly enough to reach a converged state, especially if results from a previous frame are available. Besides a fast convergence, quantitative experiments with synthetic data have been conducted. It could be shown that depending on the selected hypothesis representation, highly precise results can be achieved. Simultaneously, the comparison with two state-of-the-art stereo algorithms has shown that the IPS results are of comparable quality while IPS exhibits a superior computational performance. Likewise, a multi-view fusion of the IPS results has led to a comparable quality of the 3D as with a popular off-line multi-view work-flow that takes 20 to 30 minutes for a single frame. Finally, the proposed combination with IBVH results has been experimentally evaluated on challenging datasets with conferees that wear almost completely homogeneous black clothes. Without the 3D data from IBVH, large areas within the clothings could not be consistently reconstructed, while the combination of both algorithms drastically mitigates this shortcoming.

The presented novel photometric image adjustment has been targeting on two objectives. First, the reduction of the computational load for 3D analysis by enabling for cheaper similarity measures, and second, the preparation of eye contact rendering as the object is textured from multiple views. The matching of photometric properties between images is coupled to a geometric image registration. In this way, the quality of photometric alignment is directly related with the quality of 3D estimation. An alternating optimization procedure has been proposed for optimizing the color registration and the geometric registration rotationally until convergence. The underlying color transformation can be chosen arbitrarily according to the application requirements. The presented mathematical optimization is formulated generically in order to allow for a substitution of user-defined color mappings. For the purpose of simulating real camera input, the evaluation of the photometric adjustment has been conducted on completely unprocessed raw camera data. Based on an elementary affine RGB mapping, both objectives could be accomplished. The transformed images match from a qualitative perspective, and the 3D results for a computationally cheap SAD

are almost identical to those that are computed with a costly NCC.

The eye contact views are rendered on basis of the estimated 3D data. In contrast to other works, the virtual *eye contact camera* is not manually predefined or computed once and fixed afterwards. Instead, a novel algorithm for the automatic calibration of the *eye contact camera* has been presented. The goal has been threefold. First, user interaction should be avoided. Second, there should be no single *sweet spot* for eye contact provision. And third, in order to simulate a natural conversation situation, a conferee should be able to circumnavigate a remote chat partner to some extent while moving around. As a prerequisite, the geometrical constraints for eye contact provision have been discussed in detail. Once during the setup of the video communication system, the relative display position with respect to the cameras needs to be identified. This information, together with the 3D eye positions of the conferees, allows for the mapping of both communication sides into a common coordinate system. The mapping is computed based on the identification and straightening of partial lines of sight for both communication sides. Once the initial line of sight is found, a continuous line of sight update enables for the update of the *eye contact cameras*. The rendering results for video communication datasets illustrate the eye contact quality, the contrast between a fixed *sweet spot* and the proposed approach, and the circumnavigation of a remote conferee.

Outlook In recent years massively parallel hardware architectures have emerged and their computational power have grown exponentially. Until today, a huge amount of the performance improvements are achieved by massively increasing the number of parallel processing units. The PC hardware in appendix A that was used for the latest demonstrator setup can be considered as an exceptional configuration for a consumer solution. As the algorithms in this thesis are designed to be easily scalable in case additional processing units are provided, in a few years, all the processing might be performed on a single graphics card. At the same time, it can be thought of high-end solutions consisting of PC clusters of today's hardware with fast interconnections. Such configurations could instantaneously provide higher resolutions and frame rates. But also from an algorithmic point of view, as discussed in section 3.2.3.6, a greater computational power allows for an immediate improvement of results by simply changing the configuration parameters. Simultaneously, it would be possible to further improve the outcome of 3D processing by using more complex hypotheses representations, dynamically computed neighborhoods for hypotheses propagation, adding explicit scene flow constraints for 3D estimation, or simply including more cameras to prevent occlusions for very twisted movements of the conferees.

However, the presented photometric alignment and the 3D estimation algorithms are not restricted to video communication applications. As an arbitrary number of views can be processed, complete rooms could be captured in 3D and rendered from any viewpoint. While such an approach could be still used for general purpose telepresence setups, other applications emerge immediately. For movie production, the 3D capturing of film location or human bodies would be a valuable source for free viewpoint rendering and the addition of special effects during post-production. But also regarding recently emerged virtual reality headsets like the Oculus Rift, the required 3D input could be captured in this way.

A. Experimental Demonstrator Setup

In the course of this thesis, all presented algorithms were part of different live setups for public exhibitions and lab demonstrators. This section provides information about the cameras that were used for the recording of the datasets in appendix B and the computer hardware that was used for serving the latest lab demonstration (figure A.1).

A.1 Cameras

Three types of cameras were used for recording video communication datasets. High resolution from XIMEA and conventional HD cameras manufactured by Basler and AVT respectively. A comprehensive listing of the key properties of these cameras is provided in table A.1.

	Sensor	**Resolution**	**Frame rate**
Ximea CB200CG-CM	CMOSIS CMV20000	5120×3840	33fps
Basler acA2000-50gc	CMOSIS CMV2000	2046×1086	50fps
AVT Pike F-210C	Kodak KAI-2093	1920×1080	30fps

Table A.1: Cameras that were used within this thesis and their key characteristics.

A.2 Computer Hardware

The whole algorithmic processing for the latest lab demonstrator setup as shown in figure A.1 is performed on a dual socket machine with two Intel Xeon E5-2660v3@2.6GHz CPUs. Each CPU provides ten physical cores. In combination with the Intel Hyper-Threading technology, the PC exhibits 40 logical CPU cores in total. Complementarily, the system is equipped with eight NVIDIA Titan X graphics cards for multi-GPU processing. Each of those GPUs consists of 3072 processing cores and 12GB of graphics memory.

A.3 Public Presentations and Lab Demonstrators

During the formation of this thesis, the presented algorithms were part of different video communication demonstrator setups. In the following, a listing of these systems is provided in reverse chronological order.

A.1 Latest lab setup from 2016. Due to the large screen, between eight and sixteen cameras were used for 3D processing. The majority of results that are presented in this thesis are related to this configuration.

A.2 The previous lab setup from 2011. The smaller screen required only three cameras. In this thesis, recordings from this setup served for experiments regarding trifocal 3D processing.

A.3 Presentation of the trifocal setup at CeBit 2010 in Hannover. A video of this exhibition is available at *heise online* [Vid11].

A.4 First public presentation of the trifocal setup at the 3D Stereo Media 2009 in Liège.

A.5 Experimental installation for the evaluation of different camera configurations in the course of the FP7 project *3D Presence*.

Figure A.1: Lab setup 2015-2016, Fraunhofer Heinrich Hertz Institute in Berlin.

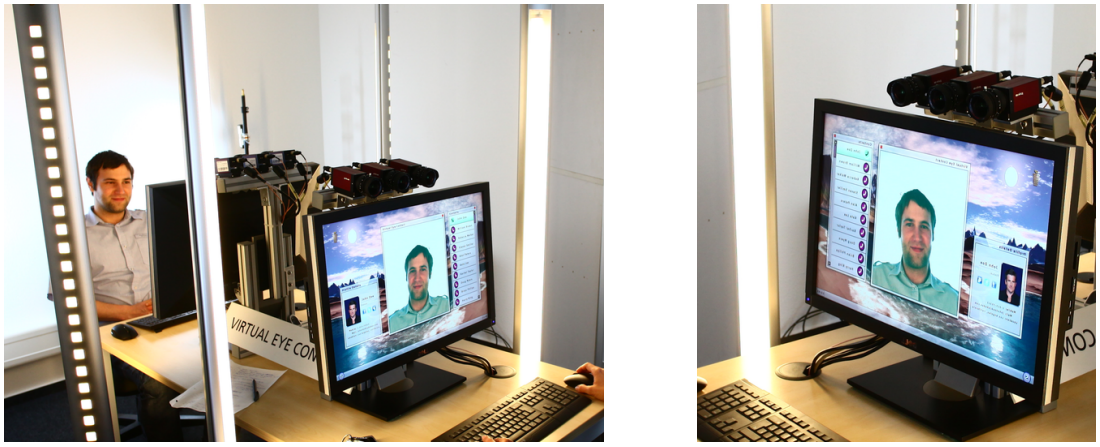


Figure A.2: Lab setup 2011-2014, Fraunhofer Heinrich Hertz Institute in Berlin.

Figure A.3: CeBit 2010, Hannover. **Top:** Booth of the Fraunhofer Society. **Bottom:** Closeup of the recorded person. The view on the left shows the original image and the view on the right the virtual eye contact perspective.

Figure A.4: 3D Stereo Media 2009, Liège. Booth of the Fraunhofer Heinrich Hertz Institute.

Figure A.5: Lab setup for the FP7 project *3D Presence* 2008, Fraunhofer Heinrich Hertz Institute in Berlin.

B. Datasets

Video communication datasets with different persons and scene dynamics and varying camera configurations have been recorded in order to evaluate the developed algorithms. A brief overview to these datasets is provided in table B.1. Corresponding example images for each view are illustrated in the referred figures. The datasets have been segmented with a state-of-the-art foreground background segmentation algorithm. All presented datasets were recorded with the demonstrator setups as described in appendix A.

Name	Views	Resolution	FPS	Length	Sample	Camera
Niklas	16	1920×1080	50	500	figure B.1	Basler acA2000-50gc
David	16	1920×1080	50	500	figure B.2	Basler acA2000-50gc
Sylvain	16	1920×1080	50	500	figure B.3	Basler acA2000-50gc
Marcus	16	1920×1080	50	500	figure B.4	Basler acA2000-50gc
Paul	16	1920×1080	50	500	figure B.5	Basler acA2000-50gc
Oliver2	16	1920×1080	50	500	figure B.6	Basler acA2000-50gc
SaschaHR	2	5120×3840	25	800	figure B.7	Ximea CB200CG-CM
RonnyHR	2	5120×3840	25	800	figure B.8	Ximea CB200CG-CM
HannesHR	2	5120×3840	25	800	figure B.9	Ximea CB200CG-CM
JohannesHR	2	5120×3840	25	800	figure B.10	Ximea CB200CG-CM
Oliver1	3	1920×1080	25	1297	figure B.11	AVT Pike F-210C

Table B.1: Overview to evaluated video communication datasets.

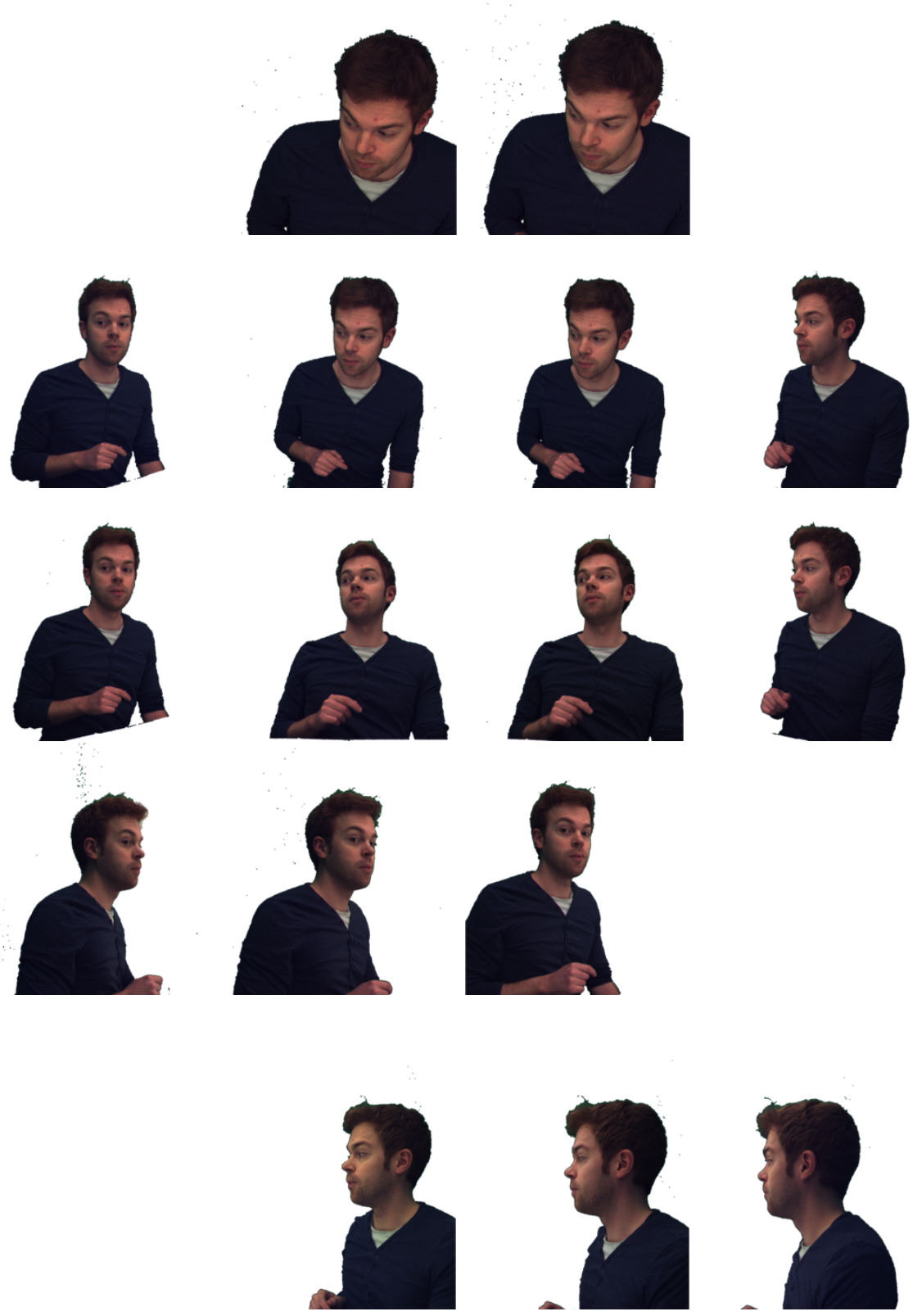


Figure B.1: Examples for all sixteen views of the `Niklas` dataset.

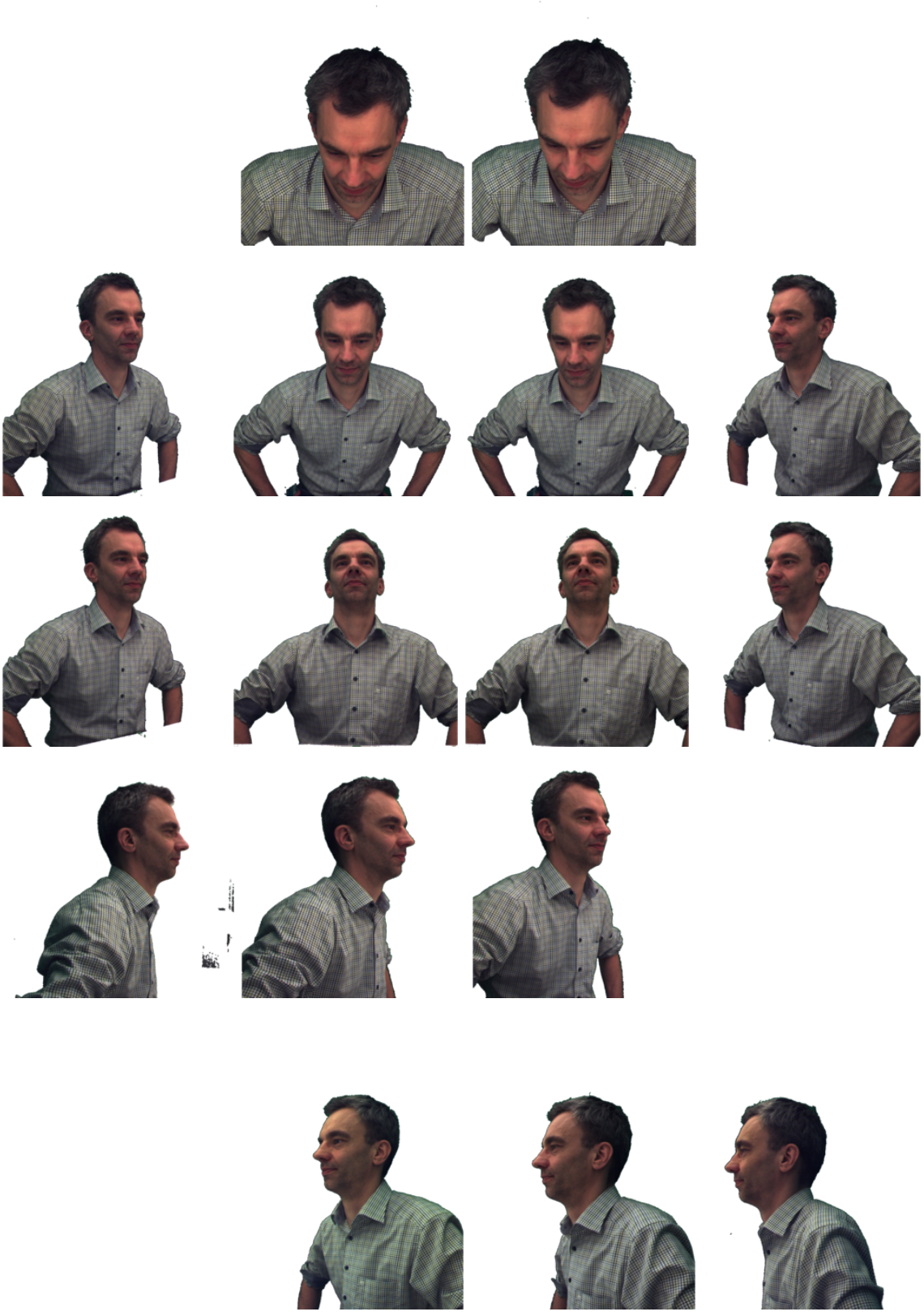


Figure B.2: Examples for all sixteen views of the `David` dataset.

Figure B.3: Examples for all sixteen views of the `Sylvain` dataset.

Figure B.4: Examples for all sixteen views of the `Marcus` dataset.

Figure B.5: Examples for all sixteen views of the `Paul` dataset.

Figure B.6: Examples for all sixteen views of the `Oliver2` dataset.

Figure B.7: Examples for both views of the `SaschaHR` dataset.

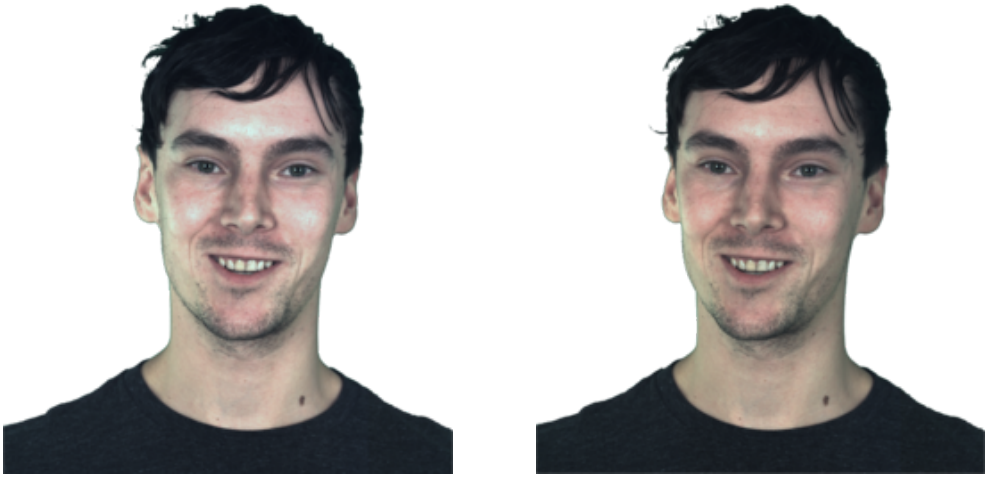


Figure B.8: Examples for both views of the `RonnyHR` dataset.

Figure B.9: Examples for both views of the `HannesHR` dataset.

Figure B.10: Examples for both views of the `JohannesHR` dataset.

Figure B.11: Examples for all three views of the `Oliver1` dataset.

References

[13a] Hybrid Recursive Analysis of Spatio-Temporal Objects. WO2013079413 A1. N. Atzpadin, W. Waizenegger, O. Schreer, I. Feldmann, P. Kauff, and P. Eisert, 2013. 5, 21

[13b] Joint Geometric and Photometric Multiview Image Registration. WO2013045651 A1. W. Waizenegger, I. Feldmann, N. Atzpadin, O. Schreer, and P. Eisert, 2013. 5, 78

[13c] View Rendering for the Provision of Virtual Eye Contact using Special Geometric Constraints in Combination with Eye-Tracking. WO2013079607 A1. N. Atzpadin, I. Feldmann, P. Kauff, O. Schreer, and W. Waizenegger, 2013. 5, 90

[17] Apparatus and Method for Performing 3D Estimation Based on Locally Determined 3D Information Hypotheses. WO/2017/207647. W. Waizenegger, O. Schreer, I. Feldmann, P. Eisert, and P. Kauff, 2017. 5

[Arm66] L. Armijo. Minimization of Functions Having Lipschitz Continuous First Partial Derivatives. *Pacific Journal of Mathematics*, 16(1):1–3, 1966. 44, 81

[AV89] N. Ahuja and J. Veenstra. Generating Octrees from Object Silhouettes in Orthographic Views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(2):137–149, 1989. 13

[Ban+10] C. Banz, S. Hesselbarth, H. Flatt, H. Blume, and P. Pirsch. Real-Time Stereo Vision System using Semi-Global Matching Disparity Estimation: Architecture and FPGA-Implementation. In *Proc. Int. Conf. on Embedded Computer Systems (SAMOS)*, pages 93–101, 2010. 15

[Bar+09] C. Barnes, E. Shechtman, A. Finkelstein, and D. Goldman. PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing. *ACM Transactions on Graphics*, 28(3):24, 2009. 15

[Bar08] A. Bartoli. Groupwise Geometric and Photometric Direct Image Registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(12):2098–2108, 2008. 17

[Bau74] B. G. Baumgart. Geometric Modeling for Computer Vision. PhD thesis, Stanford University, 1974. 13

[BBH08] D. Bradley, T. Boubekeur, and W. Heidrich. Accurate Multi-View Reconstruction using Robust Binocular Stereo and Surface Meshing. In *Proc. Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008. 20

[BCP11] H. Baker, N. L. Chang, and A. Paruchuri. Capture and Display for Live Immersive 3D Entertainment. In *Proc. ACM Int. Conf. on Multimedia (MM)*, pages 1069–1072, 2011. 33

[Bes+14] F. Besse, C. Rother, A. Fitzgibbon, and J. Kautz. PMBP: PatchMatch Belief Propagation for Correspondence Field Estimation. *International Journal of Computer Vision*, 110(1):2–13, 2014. 15, 16

[BF82] S. T. Barnard and M. A. Fischler. Computational Stereo. *ACM Computing Surveys*, 14(4):553–572, 1982. 14

[BMB06] Y. Bondareva, L. Meesters, and D. Bouwhuis. *Eye Contact as a Determinant of Social Presence in Video Communication.* Information Gatekeepers Inc. 2006. 1

[Bri+09] T. Brick, J. Spies, B.-J. Theobald, I. Matthews, and S. Boker. High-Presence, Low-Bandwidth, Apparent 3D Video-Conferencing with a Single Camera. In *Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS)*, pages 308–311, 2009. 11

[BRR11] M. Bleyer, C. Rhemann, and C. Rother. PatchMatch Stereo-Stereo Matching with Slanted Support Windows. In *Proc. British Machine Vision Conf. (BMVC)*, pages 1–11, 2011. 15

[Bud12] M. Buder. Dense Real-Time Stereo Matching using Memory Efficient Semi-Global-Matching Variant Based on FPGAs. In *Proc. SPIE 8437, Real-Time Image and Video Processing*, 2012. 15

[Bue+99] C. Buehler, W. Matusik, L. McMillan, and S. Gortler. Creating and Rendering Image-Based Visual Hulls. Technical report, Massachusetts Institute of Technology, 1999. 13, 21, 22

[CE10] J. Civit and O. Escoda. Robust Foreground Segmentation for GPU Architecture in an Immersive 3D Videoconferencing System. In *Int. Workshop on Multimedia Signal Processing (MMSP)*, pages 75–80, 2010. 20

[Che+10] G. Chen, H. Su, J. Jiang, and W. Wu. Safe Polyhedral Visual Hulls. In *Proc. Int. Conf. on Advances in Multimedia Modeling (MMM)*, pages 35–44, 2010. 13

[CKJ02] T.-J. Cham, S. Krishnamoorthy, and M. Jones. Analogous View Transfer for Gaze Correction in Video Sequences. In *Int. Conf. on Control, Automation, Robotics and Vision (ICARCV)*, pages 1415–1420, 2002. 10

[CM09] J. Civit and T. Montserrat. Eye Gaze Correction to Guarantee Eye Contact in Videoconferencing. *IEEE Latin America Transactions*, 7(3):405–409, 2009. 9

[Col96] R. T. Collins. A Space-Sweep Approach to True Multi-Image Matching. In *Proc. Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 1996. 14, 33

[Coo+03] E. Cooke, I. Feldmann, P. Kauff, and O. Schreer. Multi-View Synthesis for an Extendable Teleconferencing System. In *Proc. of Picture Coding Symposium (PCS)*, pages 199–204, 2003. 19

[Cox+92] I. J. Cox, S. Hingorani, B. M. Maggs, and S. B. Rao. Stereo Without Disparity Gradient Smoothing: A Bayesian Sensor Fusion Solution. In *Proc. British Machine Vision Conf. (BMVC)*, pages 337–346, 1992. 8, 14

[Cri+03] A. Criminisi, J. Shotton, A. Blake, and P. Torr. Gaze Manipulation for One-To-One Teleconferencing. In *Proc. Int. Conf. on Computer Vision (ICCV)*, 2003. 8, 14, 18

[CT12] F. Calakli and G. Taubin. SSD-C: Smooth Signed Distance Colored Surface Reconstruction. *Expanding the Frontiers of Visual Analytics and Visualization*, pages 323–338. Springer London, 2012. 72

[Div+10] O. Divorra Escoda, J. Civit, F. Zuo, H. Belt, I. Feldmann, O. Schreer, E. Yellin, W. Ijsselsteijn, R. van Eijk, D. Espinola, P. Hagendorf, W. Waizenegger, and R. Braspenning. Towards 3D-Aware Telepresence: Working on Technologies Behind the Scene. In *Proc. ACM conference on Computer Supported Cooperative Work (CSCW)*, 2010. 14

[DN09] C. Doutre and P. Nasiopoulos. Color Correction Preprocessing for Multiview Video Coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 19(9):1400–1406, 2009. 17

[Dou+12] M. Dou, Y. Shi, J. Frahm, H. Fuchs, B. Mauchly, and M. Marathe. Room-Sized Informal Telepresence System. In *Proc. of Virtual Reality Short Papers and Posters (VRW)*, pages 15–18, 2012. 9

[DR11] T. Duckworth and D. J. Roberts. Accelerated Polyhedral Visual Hulls using OpenCL. In *Proc. of Virtual Reality Conference (VR)*, pages 203–204, 2011. 13

[Dum+08] M. Dumont, S. Maesen, S. Rogmans, and P. Bekaert. A Prototype for Practical Eye-Gaze Corrected Video Chat on Graphics Hardware. In *Proc. Int. Conf. on Signal Processing and Multimedia Applications (SIGMAP)*, pages 236–243, 2008. 9, 14, 18

[Dum+09] M. Dumont, S. Rogmans, S. Maesen, and P. Bekaert. Optimized Two-Party Video Chat with Restored Eye Contact Using Graphics Hardware. *e-Business and Telecommunications*, pages 358–372. Springer Berlin Heidelberg, 2009. 9, 14

[Dum+10] M. Dumont, S. Rogmans, G. Lafruit, and P. Bekaert. Immersive Teleconferencing with Natural 3D Stereoscopic Eye Contact using GPU Computing. In *Proc. of 3D Stereo Media*, 2010. 9, 14, 18

[DVE] DVE Telepresense, Huddle 70. URL: `http://www.dvetelepresence.com/dve_huddle.php` (visited on 08/02/2016). 12

[Ebe+14] S. Ebel, W. Waizenegger, M. Reinhardt, O. Schreer, and I. Feldmann. Visibility-Driven Patch Group Generation. In *Proc. Int. Conf. on 3D Imaging (IC3D)*, pages 1–8, 2014. 20, 21, 72

[EE13] N. Einecke and J. Eggert. Stereo Image Warping for Improved Depth Estimation of Road Surfaces. In *Proc. of Intelligent Vehicles Symposium (IV)*, pages 189–194, 2013. 15

[EH08] I. Ernst and H. Hirschmüller. Mutual Information Based Semi-Global Stereo Matching on the GPU. In *Proc. Int. Symposium on Advances in Visual Computing (ISVC)*, pages 228–239, 2008. 15

[ER06] P. Eisert and J. Rurainsky. Geometry-Assisted Image-Based Rendering for Facial Analysis and Synthesis. *Signal Processing: Image Communication*, 21(6):493–505, 2006. 10

[Far+14] H. S. Faridul, T. Pouli, C. Chamaret, J. Stauder, A. Tremeau, and E. Reinhard. A Survey of Color Mapping and its Applications. In *Eurographics State of the Art Reports*, 2014. 16

[FB03] J.-S. Franco and E. Boyer. Exact Polyhedral Visual Hulls. In *Proc. British Machine Vision Conf. (BMVC)*, pages 32.1–32.10, 2003. 13

[FB09] J.-S. Franco and E. Boyer. Efficient Polyhedral Modeling from Silhouettes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(3):414–427, 2009. 13

[Fel+09] I. Feldmann, N. Atzpadin, O. Schreer, J.-C. Pujol-Acolado, J. Landabaso, and O. Escoda. Multi-View Depth Estimation Based on Visual-Hull Enhanced Hybrid Recursive Matching for 3D Video Conference Systems. In *Proc. Int. Conf. on Image Processing (ICIP)*, pages 745–748, 2009. 14

[Fel+10] I. Feldmann, W. Waizenegger, N. Atzpadin, and O. Schreer. Real-Time Depth Estimation for Immersive 3D Videoconferencing. In *Proc. of 3DTV-Conference (3DTV-CON)*, pages 1–4, 2010. 14

[FL15a] S. A. Fezza and M.-C. Larabi. Color Calibration of Multi-View Video plus Depth for Advanced 3D Video. *Signal, Image and Video Processing*, 9(1):177–191, 2015. 17

[FL15b] S. A. Fezza and M.-C. Larabi. Color Correction for Stereo and Multi-View Coding. *Color Image and Video Enhancement*, pages 291–314. Springer International Publishing, 2015. 16

[Fle+14] C. Fleury, T. Popa, T.-J. Cham, and H. Fuchs. Merging Live and Pre-Captured Data to Support Full 3D Head Reconstruction for Telepresence. In *Eurographics Short Papers*, pages 9–12, 2014. 11

[FP07] Y. Furukawa and J. Ponce. Accurate, Dense, and Robust Multi-View Stereopsis. In *Proc. Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2007. 17, 33, 72

[FP10] Y. Furukawa and J. Ponce. Accurate, Dense, and Robust Multiview Stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8):1362–1376, 2010. 19, 33, 72

[FP708] FP7-ICT, 3DPresence. URL: `http://cordis.europa.eu/project/rcn/85538_en.html` (visited on 06/02/2017). 12

[FSC98] G. D. Finlayson, B. Schiele, and J. L. Crowley. Comprehensive Colour Image Normalization. In *Proc. Europ. Conf. on Computer Vision (ECCV)*, pages 475–490, 1998. 16

[Fur] Y. Furukawa, PMVS2. URL: `http://www.di.ens.fr/pmvs/` (visited on 11/24/2016). 72

[Gem+00] J. Gemmell, K. Toyama, C. Zitnick, T. Kang, and S. Seitz. Gaze Awareness for Video-Conferencing: A Software Approach. *IEEE Multimedia*, 7(4):26–35, 2000. 10, 18

[GN03] M. D. Grossberg and S. K. Nayar. Determining the Camera Response from Images: What Is Knowable? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(11):1455–1467, 2003. 16

[Gro+03] M. Gross, S. Wuermlin, M. Naef, E. Lamboray, C. Spagno, A. Kunz, E. Koller-Meier, T. Svoboda, L. V. Gool, S. Lang, K. Strehlke, A. V. Moere, and O. Staadt. Blue-C: A Spatially Immersive Display and 3D Video Portal for Telepresence. In *Proc. of ACM SIGGRAPH*, pages 819–827, 2003. 8

[GSD03] K. Grauman, G. Shakhnarovich, and T. Darrell. A Bayesian Approach to Image-Based Visual Hull Reconstruction. In *Proc. Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 187–194, 2003. 13

[GSF14] S. Gehrig, N. Schneider, and U. Franke. Exploiting Traffic Scene Disparity Statistics for Stereo Vision. In *Proc. Int. Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 688–695, 2014. 15

[GTZ99] J. Gemmell, K. Toyama, and C. L. Zitnick. Manipulation of Video Eye Gaze and Head Orientation for Video Teleconferencing. Technical report, Microsoft Research, 1999. 10

[GZ02] J. Gemmell and D. Zhu. Implementing Gaze-Corrected Videoconferencing. In *Communications, Internet, and Information Technology*, pages 382–387, 2002. 10, 18

[Hau+12] S. Hauswiesner, R. Khlebnikov, M. Steinberger, M. Straka, and G. Reitmayr. Multi-GPU Image-Based Visual Hull Rendering. In *Proc. of the Eurographics Symposium on Parallel Graphics and Visualization*, pages 119–128, 2012. 13

[HBE12] H. Hirschmüller, M. Buder, and I. Ernst. Memory Efficient Semi-Global Matching. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 3(1):371–376, 2012. 15

[HE09] A. Hilsmann and P. Eisert. Joint Estimation of Deformable Motion and Photometric Parameters in Single View Video. In *Proc. Int. Conf. on Computer Vision Workshops (ICCV Workshops)*, pages 390–397, 2009. 17

[Hei+13] P. Heise, S. Klose, B. Jensen, and A. Knoll. PM-Huber: PatchMatch with Huber Regularization for Stereo Matching. In *Proc. Int. Conf. on Computer Vision (ICCV)*, pages 2360–2367, 2013. 15

[Hir05] H. Hirschmüller. Accurate and Efficient Stereo Processing by Semi-Global Matching and Mutual Information. In *Proc. Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 807–814, 2005. 15, 65

[Hir06] H. Hirschmüller. Stereo Vision in Structured Environments by Consistent Semi-Global Matching. In *Proc. Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2386–2393, 2006. 15

[Hir08] H. Hirschmüller. Stereo Processing by Semiglobal Matching and Mutual Information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):328–341, 2008. 15

[HK12] S. Hermann and R. Klette. Iterative Semi-Global Matching for Robust Driver Assistance Systems. In *Proc. Asian Conference on Computer Vision (ACCV)*, pages 465–478, 2012. 15

[HLL08] Y. S. Heo, K. M. Lee, and S. U. Lee. Illumination and Camera Invariant Stereo Matching. In *Proc. Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008. 77

[Hoe12] A. V. D. Hoest. Eye Contact in Leisure Video Conferencing. Master thesis, Gjøvik University College, 2012. 1

[HOP14] D. Honegger, H. Oleynikova, and M. Pollefeys. Real-Time and Low Latency Embedded Computer Vision Hardware Based on a Combination of FPGA and Mobile CPU. In *Proc. Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 4930–4935, 2014. 15

[HS07] H. Hirschmüller and D. Scharstein. Evaluation of Cost Functions for Stereo Matching. In *Proc. Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007. 16, 77

[HS09] H. Hirschmüller and D. Scharstein. Evaluation of Stereo Matching Costs on Images with Radiometric Differences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(9):1582–1599, 2009. 77

[HZ04] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press. 2nd ed. 2004. 96

[Its15] Itseez, Open Source Computer Vision Library. URL: `https://github.com/itseez/opencv` (visited on 01/27/2016). 66

[IW05] A. Ilie and G. Welch. Ensuring Color Consistency Across Multiple Cameras. In *Proc. Int. Conf. on Computer Vision (ICCV)*, pages 1268–1275, 2005. 16

[JD02] J. Jerald and M. Daily. Eye Gaze Correction for Videoconferencing. In *Proc. of the symposium on Eye tracking research & applications (ETRA)*, pages 77–81, 2002. 10, 18

[Jon+09] A. Jones, M. Lang, G. Fyffe, X. Yu, J. Busch, I. McDowall, M. Bolas, and P. Debevec. Achieving Eye Contact in a One-To-Many 3D Video Teleconferencing System. *ACM Transactions on Graphics*, 28(3):64:1–64:8, 2009. 11

[Kaz] M. Kazhdan, Screened Poisson Surface (and Smoothed Signed Distance) Reconstruction (V9.01). URL: `http://www.cs.jhu.edu/~misha/Code/PoissonRecon/Version9.01/` (visited on 11/24/2016). 72

[KBH06] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson Surface Reconstruction. In *Proc. of the fourth Eurographics symposium on Geometry processing (SGP)*, pages 61–70, 2006. 72

[KE06] C. Küblbeck and A. Ernst. Face Detection and Tracking in Video Sequences using the Modified Census Transformation. *Image and Vision Computing*, 24(6):564–572, 2006. 96

[Ket+10] M. Kettern, D. C. Schneider, B. Prestele, F. Zilly, and P. Eisert. Automatic Acquisition of Time-Slice Image Sequences. In *Proc. Conf. on Visual Media Production (CVMP)*, pages 40–48, 2010. 17

[KFP08] S. J. Kim, J.-M. Frahm, and M. Pollefeys. Radiometric Calibration with Illumination Change for Outdoor Scene Analysis. In *Proc. Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008. 16

[KH13] M. Kazhdan and H. Hoppe. Screened Poisson Surface Reconstruction. *ACM Transactions on Graphics*, 32(3):29.1–29.13, 2013. 72

[Kje+14] J. Kjeldskov, J. H. Smedegård, T. S. Nielsen, M. B. Skov, and J. Paay. Eye Contact over Video. In *Extended Abstracts on Human Factors in Computing Systems (CHI EA)*, pages 1561–1566, 2014. 10

[KK06] M. Kuechler and A. Kunz. HoloPort - A Device for Simultaneous Video and Data Conferencing Featuring Gaze Awareness. In *Proc. of Virtual Reality Conference (VR)*, pages 81–88, 2006. 12

[KP08] S. J. Kim and M. Pollefeys. Robust Radiometric Calibration and Vignetting Correction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(4):562–576, 2008. 16

[KS02] P. Kauff and O. Schreer. An Immersive 3D Video-Conferencing System using Shared Virtual Team User Environments. In *Proc. Int. Conf. on Collaborative Virtual Environments (CVE)*, pages 105–112, 2002. 8, 14

[KSO01] P. Kauff, O. Schreer, and J.-R. Ohm. An Universal Algorithm for Real-Time Estimation of Dense Displacement Vector Fields. In *Proc. of Int. Conf. on Media Futures*, 2001. 14

[Kur+13] G. Kurillo, H. Baker, Z. Li, and R. Bajcsy. Geometric and Color Calibration of Multiview Panoramic Cameras for Life-Size 3D Immersive Video. In *Proc. Int. Conf. on 3D Vision (3DV)*, pages 374–381, 2013. 17

[Kus+11] C. Kuster, T. Popa, C. Zach, C. Gotsman, and M. Gross. FreeCam: A Hybrid Camera System for Interactive Free-Viewpoint Video. In *Proc. Annual Workshop on Vision, Modeling and Visualization (VMV)*, pages 17–24, 2011. 10

[Kus+12a] C. Kuster, N. Ranieri, A. Agustina, H. Zimmer, J. Bazin, C. Sun, T. Popa, and M. Gross. Towards Next Generation 3D Teleconferencing Systems. In *Proc. of 3DTV-Conference (3DTV-CON)*, pages 1–4, 2012. 9

[Kus+12b] C. Kuster, T. Popa, J.-C. Bazin, C. Gotsman, and M. Gross. Gaze Correction for Home Video Conferencing. *ACM Transactions on Graphics*, 31(6):174.1–174.6, 2012. 11, 18

[LA09] M. I. A. Lourakis and A. A. Argyros. SBA: A Software Package for Generic Sparse Bundle Adjustment. *ACM Transactions on Mathematical Software*, 36(1):1–30, 2009. 96

[Lau94] A. Laurentini. The Visual Hull Concept for Silhouette-Based Image Understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):150–162, 1994. 13

[LBN08] A. Ladikos, S. Benhimane, and N. Navab. Efficient Visual Hull Computation for Real-Time 3D Reconstruction using CUDA. In *Proc. Int. Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1–8, 2008. 13

[LBW95] J. Liu, I. P. Beldie, and M. Wöpking. A Computational Approach To Establish Eye-Contact In Videocommunication. In *Proc. Int. Workshop on Stereoscopic and Three Dimensional Imaging (IWS3DI)*, pages 229–234, 1995. 8, 14

[LC87] W. E. Lorensen and H. E. Cline. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. *ACM SIGGRAPH Computer Graphics*, 21(4):163–169, 1987. 23

[LDX10] K. Li, Q. Dai, and W. Xu. High Quality Color Calibration for Multi-Camera Systems with an Omnidirectional Color Checker. In *Proc. Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1026–1029, 2010. 16

[LDX11] K. Li, Q. Dai, and W. Xu. Collaborative Color Calibration for Multi-Camera Systems. *Signal Processing: Image Communication*, 26(1):48–60, 2011. 16

[LFP07] S. Lazebnik, Y. Furukawa, and J. Ponce. Projective Visual Hulls. *International Journal of Computer Vision*, 74(2):137–165, 2007. 13

[LH02] B. J. Lei and E. A. Hendriks. Real-Time Multi-Step View Reconstruction for a Virtual Teleconference System. *EURASIP Journal on Applied Signal Processing*, 2002(1):1067–1087, 2002. 8, 14

[Lu+13] J. Lu, H. Yang, D. Min, and M. Do. Patch Match Filter: Efficient Edge-Aware Filtering Meets Randomized Search for Fast Correspondence Field Estimation. In *Proc. Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1854–1861, 2013. 15, 16

[Lu+15] S.-P. Lu, B. Ceulemans, A. Munteanu, and P. Schelkens. Spatio-Temporally Consistent Color and Structure Optimization for Multiview Video Color Correction. *IEEE Transactions on Multimedia*, 17(5):577–590, 2015. 17

[Mai+12] A. Maimone, J. Bidwell, K. Peng, and H. Fuchs. Enhanced Personal Autostereoscopic Telepresence System using Commodity Depth Cameras. *Computers & Graphics*, 36(7):791–807, 2012. 9, 20

[Mai+13] A. Maimone, X. Yang, N. Dierk, A. State, M. Dou, and H. Fuchs. General-Purpose Telepresence with Head-Worn Optical See-Through Displays and Projector-Based Lighting. In *Proc. of Virtual Reality Conference (VR)*, pages 23–26, 2013. 9

[Mat+00] W. Matusik, C. Buehler, R. Raskar, S. J. Gortler, and L. McMillan. Image-Based Visual Hulls. In *Proc. of the Annual Conf. on Computer Graphics and Interactive Techniques*, pages 369–374, 2000. 8, 13, 21, 22

[Mat+02a] W. Matusik, C. Buehler, L. McMillan, and S. Gortler. An Efficient Visual Hull Computation Algorithm. Technical report, Massachusetts Institute of Technology, 2002. 13, 21, 22

[Mat+02b] W. Matusik, C. Buehler, L. McMillan, and S. J. Gortler. Efficient View-Dependent Sampling of Visual Hulls. Technical report, Massachusetts Institute of Technology, 2002. 13, 21, 22

[Mat01] W. Matusik. Image-Based Visual Hulls. Master thesis, Massachusetts Institute of Technology, 2001. 21, 22

[MBM01] W. Matusik, C. Buehler, and L. McMillan. Polyhedral Visual Hulls for Real-Time Rendering. In *Proc. of the 12th Eurographics Workshop on Rendering Techniques*, pages 115–126, 2001. 13

[MBP95] L. Mühlbach, M. Bocker, and A. Prussog. Telepresence in Videocommunications: A Study on Stereoscopy and Individual Eye Contact. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37(2):290–305, 1995. 1

[MF11a] A. Maimone and H. Fuchs. A First Look at a Telepresence System with Room-Sized Real-Time 3D Capture and Large Tracked Display. In *Proc. Int. Conf. on Artifical Reality and Telexistence (ICAT)*, 2011. 9

[MF11b] A. Maimone and H. Fuchs. Encumbrance-Free Telepresence System with Real-Time 3D Capture and Display using Commodity Depth Cameras. In *Proc. Int. Symposium on Mixed and Augmented Reality (ISMAR)*, pages 137–146, 2011. 9

[MF12] A. Maimone and H. Fuchs. Real-Time Volumetric 3D Capture of Room-Sized Scenes for Telepresence. In *Proc. of 3DTV-Conference (3DTV-CON)*, pages 1–4, 2012. 9, 20

[MG15a] M. Menze and A. Geiger, KITTI Vision Benchmark Suite. URL: `http://ww.w.cvlibs.net/publications/Menze2015CVPR.pdf` (visited on 11/03/2016). 12

[MG15b] M. Menze and A. Geiger. Object Scene Flow for Autonomous Vehicles. In *Proc. Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3061–3070, 2015. 12

[Mic+13] M. Michael, J. Salmen, J. Stallkamp, and M. Schlipsing. Real-Time Stereo Vision: Optimizing Semi-Global Matching. In *Proc. of Intelligent Vehicles Symposium (IV)*, pages 1197–1202, 2013. 15, 65, 68

[Mül+09] K. Müller, A. Smolic, K. Dix, P. Merkle, P. Kauff, and T. Wiegand. View Synthesis for Advanced 3D Video Systems. *EURASIP Journal on Image and Video Processing*, 2009(1):1–11, 2009. 19

[Mur+10] D. Murayama, K. Kimura, T. Hosaka, T. Hamamoto, N. Shibuhisa, S. Tanaka, S. Sato, and S. Saito. Virtual View Image Synthesis for Eye-Contact in TV Conversation System. In *Proc. of Three-Dimensional Image Processing and Applications (3DIP)*, 2010. 19

[MZK10] M. Müller, F. Zilly, and P. Kauff. Adaptive Cross-Trilateral Depth Map Filtering. In *Proc. of 3DTV-Conference (3DTV-CON)*, pages 1–4, 2010. 101

[NC05] D. Nguyen and J. Canny. MultiView: Spatially Faithful Group Video Conferencing. In *Proc. of Conf. on Human Factors in Computing Systems (CHI)*, pages 799–808, 2005. 11

[New+11] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. KinectFusion: Real-Time Dense Surface Mapping and Tracking. In *Proc. Int. Symposium on Mixed and Augmented Reality (ISMAR)*, pages 127–136, 2011. 10, 11

[NFA88] H. Noborio, S. Fukuda, and S. Arimoto. Construction of the Octree Approximating a Three-Dimensional Object by using Multiple Views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(6):769–782, 1988. 13

[NKM10] S. Nobuhara, Y. Kimura, and T. Matsuyama. Object-Oriented Color Calibration of Multi-Viewpoint Cameras in Sparse and Convergent Arrangement. *IPSJ Transactions on Computer Vision and Applications*, 2(1):132–144, 2010. 17

[NVI14] NVIDIA, CuRAND. URL: `https://developer.nvidia.com/cuRAND` (visited on 02/27/2015). 45

[Ohm+98] J.-R. Ohm, K. Grüneberg, E. Hendriks, E. Izquierdo, D. Kaliva, M. Karl, D. Papadimatos, and A. Redert. A Realtime Hardware System for Stereoscopic Videoconferencing with Viewpoint Adaptation. *Signal Processing: Image Communication*, 14(1):147–171, 1998. 8, 14

[Oka+94] K. Okada, F. Maeda, Y. Ichikawaa, and Y. Matsushita. Multiparty Videoconferencing at Virtual Social Distance: MAJIC Design. In *Proc. ACM conference on Computer Supported Cooperative Work (CSCW)*, pages 385–393, 1994. 12

[OLC93] M. Ott, J. P. Lewis, and I. Cox. Teleconferencing Eye Contact using a Virtual Camera. In *Proc. of Conf. on Human Factors in Computing Systems (CHI)*, pages 109–110, 1993. 8, 14

[OTM96] K. Okada, S. Tanaka, and Y. Matsushita. MAJIC and DesktopMAJIC Conferencing System. In *Proc. ACM conference on Computer Supported Cooperative Work (CSCW)*, 1996. 12

[Pan+10] M. Panahpour Tehrani, A. Ishikawa, S. Sakazawa, and A. Koike. Iterative Colour Correction of Multicamera Systems using Corresponding Feature Points. *Journal of Visual Communication and Image Representation*, 21(5):377–391, 2010. 17

[Poc+08] T. Pock, T. Schoenemann, G. Graber, H. Bischof, and D. Cremers. A Convex Formulation of Continuous Multi-Label Problems. In *Proc. Europ. Conf. on Computer Vision (ECCV)*, pages 792–805, 2008. 80

[Pöl+12] M. Pölönen, J. Hakala, R. Bilcu, T. Järvenpää, J. Häkkinen, and M. Salmimaa. Color Asymmetry in 3D Imaging: Influence on the Viewing Experience. *3D Research*, 3(3):1–10, 2012. 16

[Pot87] M. Potmesil. Generating Octree Models of 3D Objects from Their Silhouettes in a Sequence of Images. *Computer Vision Graphics and Image Processing*, 40(1):1–29, 1987. 13

[QM99] B. Quante and L. Muehlbach. Eye-Contact in Multipoint Videoconferencing. In *Proc. of the 17th Int. Symposium on Human Factors in Telecommunications*, 1999. 1

[Reg+12] H. Regenbrecht, L. Müller, S. Hoermann, T. Langlotz, and A. Duenser. Implementing Eye-To-Eye Contact in Life-Sized Videoconferencing. Technical report, University of Otago, 2012. 12

[Rie+12a] C. Riechert, F. Zilly, P. Kauff, J. Güther, and R. Schäfer. Fully Automatic Stereo-To-Multiview Conversion in Autostereoscopic Displays. *The Best of IET and IBC*, 4(1):8–14, 2012. 15, 19, 65

[Rie+12b] C. Riechert, F. Zilly, M. Müller, and P. Kauff. Real-Time Disparity Estimation Using Line-Wise Hybrid Recursive Matching and Cross-Bilateral Median Up-Sampling. In *Proc. Int. Conf. on Pattern Recognition (ICPR)*, pages 3168–3171, 2012. 15, 65, 67, 101

[Rog+09a] S. Rogmans, M. Dumont, T. Cuypers, G. Lafruit, and P. Bekaert. Complexity Reduction of Real-Time Depth Scanning on Graphics Hardware. In *Proc. Int. Conf. on Computer Vision Theory and Applications (VISAPP)*, pages 547–550, 2009. 14

[Rog+09b] S. Rogmans, J. Lu, P. Bekaert, and G. Lafruit. Real-Time Stereo-Based View Synthesis Algorithms: A Unified Framework and Evaluation on Commodity GPUs. *Image Communication*, 24(1):49–64, 2009. 14

[RZK11] C. Riechert, F. Zilly, and P. Kauff. Real Time Depth Estimation using Line Recursive Matching. In *Proc. Conf. on Visual Media Production (CVMP)*, 2011. 15, 33, 65

[Sei+06a] S. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, Multi-View Stereo Benchmark. URL: `http://vision.middlebury.edu/mview/` (visited on 11/08/2016). 12, 19, 33

[Sei+06b] S. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms. In *Proc. Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 519–528, 2006. 12, 19, 33

[SLC11] J. M. Saragih, S. Lucey, and J. F. Cohn. Deformable Model Fitting by Regularized Landmark Mean-Shift. *International Journal of Computer Vision*, 91(2):200–215, 2011. 11

[Soa+07] L. Soares, C. Merrier, B. Raffin, and J.-L. Roch. Parallel Adaptive Octree Carving for Real-Time 3D Modeling. In *Proc. of Virtual Reality Conference (VR)*, pages 273–274, 2007. 13

[SR11] F. Solina and R. Ravnik. Fixing Missing Eye-Contact in Video Conferencing Systems. In *Proc. Int. Conf. on Information Technology Interfaces (ITI)*, pages 233–236, 2011. 10

[SS02a] D. Scharstein and R. Szeliski. A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. *International Journal of Computer Vision*, 47(1):7–42, 2002. 12, 14

[SS02b] D. Scharstein and R. Szeliski, Stereo Benchmark. URL: `http://vision.middlebury.edu/stereo/` (visited on 04/11/2016). 12, 14, 16

[Str+08a] C. Strecha, W. v. Hansen, L. V. Gool, P. Fua, and U. Thoennessen, Multi-View Stereo Benchmark. URL: `http://cvlabwww.epfl.ch/data/multiview/` (visited on 11/08/2016). 12

[Str+08b] C. Strecha, W. v. Hansen, L. V. Gool, P. Fua, and U. Thoennessen. On Benchmarking Camera Calibration and Multi-View Stereo for High Resolution Imagery. In *Proc. Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008. 12

[Sze93] R. Szeliski. Rapid Octree Construction from Image Sequences. *CVGIP: Image Understanding*, 58(1):23–32, 1993. 13

[TR00] M. J. Taylor and S. M. Rowe. Gaze Communication using Semantically Consistent Spaces. In *Proc. of Conf. on Human Factors in Computing Systems (CHI)*, pages 400–407, 2000. 11

[Tsa+04] Y.-P. Tsai, C.-C. Kao, Y.-P. Hung, and Z.-C. Shih. Real-Time Software Method for Preserving Eye Contact in Video Conferencing. *Journal of Information Science and Engineering*, 20(5):1001–1017, 2004. 8, 14

[Ver+03] R. Vertegaal, I. Weevers, C. Sohn, and C. Cheung. GAZE-2: Conveying Eye Contact in Group Video Conferencing using Eye-Controlled Camera Direction. In *Proc. of Conf. on Human Factors in Computing Systems (CHI)*, 2003. 12, 18

[Vid11] heise Video, Virtual-Eye-Contact-Engine. URL: `http://www.heise.de/video/thema/?thema=Virtual-Eye-Contact-Engine` (visited on 11/24/2016). 110

[VWS02] R. Vertegaal, I. Weevers, and C. Sohn. GAZE-2: An Attentive Video Conferencing System. In *Extended Abstracts on Human Factors in Computing Systems (CHI EA)*, pages 736–737, 2002. 12

[Wai+09] W. Waizenegger, I. Feldmann, P. Eisert, and P. Kauff. Parallel High Resolution Real-Time Visual Hull on GPU. In *Proc. Int. Conf. on Image Processing (ICIP)*, pages 4245–4248, 2009. 5, 13, 21

[Wai+11] W. Waizenegger, N. Atzpadin, O. Schreer, and I. Feldmann. Patch-Sweeping with Robust Prior for High Precision Depth Estimation in Real-Time Systems. In *Proc. Int. Conf. on Image Processing (ICIP)*, pages 881–884, 2011. 5, 21

[Wai+12] W. Waizenegger, N. Atzpadin, O. Schreer, I. Feldmann, and P. Eisert. Model Based 3D Gaze Estimation for Provision of Virtual Eye Contact. In *Proc. Int. Conf. on Image Processing (ICIP)*, pages 1973–1976, 2012. 5, 90

[Wai+13] W. Waizenegger, I. Feldmann, O. Schreer, and P. Eisert. Scene Flow Constrained Multi-Prior Patch-Sweeping for Real-Time Upper Body 3D Reconstruction. In *Proc. Int. Conf. on Image Processing (ICIP)*, pages 2086–2090, 2013. 5, 21

[Wai+16] W. Waizenegger, I. Feldmann, O. Schreer, P. Kauff, and P. Eisert. Real-Time 3D Body Reconstruction for Immersive TV. In *Proc. Int. Conf. on Image Processing (ICIP)*, pages 360–364, 2016. 5

[Wai09] W. Waizenegger. Multi-View Depth Estimation Based on Combination of Visual-Hull and Hybrid Recursive Matching. 3D Processing Workshop, BBC Research & Development, England, 2009. 5

[Wel+05] G. Welch, H. Fuchs, B. Cairns, K. Mayer-Patel, D. H. Sonnenwald, R. Yang, A. State, H. Towles, A. Ilie, M. Noland, V. Noel, and H. Yang. Improving, Expanding and Extending 3D Telepresence. In *Proc. Int. Workshop on Advanced Information Processing for Ubiquitous Networks/Int. Conf. on Artifical Reality and Telexistence (ICAT)*, 2005. 8, 14

[WFE11] W. Waizenegger, I. Feldmann, and P. Eisert. Depth Driven Photometric and Geometric Image Registration for Real-Time Stereo Systems. In *Proc. Annual Workshop on Vision, Modeling and Visualization (VMV)*, pages 25–32, 2011. 5, 17, 78

[WFS11] W. Waizenegger, I. Feldmann, and O. Schreer. Real-Time Patch Sweeping for High-Quality Depth Estimation in 3D Videoconferencing Applications. In *Proc. SPIE 7871, Real-Time Image and Video Processing*, 2011. 5, 21

[WSW10] Q. Wang, X. Sun, and Z. Wang. A Robust Algorithm for Color Correction Between Two Stereo Images. In *Proc. Asian Conference on Computer Vision (ACCV)*, pages 405–416, 2010. 17

[WTM06] X. Wu, O. Takizawa, and T. Matsuyama. Parallel Pipeline Volume Intersection for Real-Time 3D Shape Reconstruction on a PC Cluster. In *Proc. Int. Conf. on Computer Vision Systems (ICVS)*, 2006. 13

[WYD07] L. Wang, R. Yang, and J. E. Davis. BRDF Invariant Stereo using Light Transport Constancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(9):1616–1626, 2007. 77

[XM10] W. Xu and J. Mulligan. Performance Evaluation of Color Correction Approaches for Automatic Multi-View Image and Video Stitching. In *Proc. Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 263–270, 2010. 16

[Xu+15] S. Xu, F. Zhang, X. He, X. Shen, and X. Zhang. PM-PM: PatchMatch With Potts Model for Object Segmentation and Stereo Matching. *IEEE Transactions on Image Processing*, 24(7):2182–2196, 2015. 15, 16

[Yan+04] R. Yang, M. Pollefeys, H. Yang, and G. Welch. A Unified Approach to Real-Time, Multi-Resolution, Multi-Baseline 2D View Synthesis and 3D Depth Estimation using Commodity Graphics Hardware. *International Journal of Image and Graphics*, 04(4):627–651, 2004. 9, 14, 19

[Yan03] R. Yang. View-Dependent Pixel Coloring: A Physically-Based Approach for Two-Dimensional View Synthesis. PhD thesis, The University of North Carolina at Chapel Hill, 2003. 9

[Yip05] B. Yip. Face and Eye Rectification in Video Conference using Affine Transform. In *Proc. Int. Conf. on Image Processing (ICIP)*, pages 513–516, 2005. 10

[YO08] K. Yamamoto and R. Oi. Color Correction for Multi-View Video using Energy Minimization of View Networks. *International Journal of Automation and Computing*, 5(3):234–245, 2008. 17

[Yve+14] P. Yver, S. Kramm, A. Bensrhair, and E. Azzam. A Novel Global Color Correction Method for 3D Content. In *Proc. SPIE 9013, Three-Dimensional Image Processing, Measurement, and Applications (3DIPM)*, 2014. 17

[YZ02] R. Yang and Z. Zhang. Eye Gaze Correction with Stereovision for Video-Teleconferencing. In *Proc. Europ. Conf. on Computer Vision (ECCV)*, pages 479–494, 2002. 10

[YZ04] R. Yang and Z. Zhang. Eye Gaze Correction with Stereovision for Video-Teleconferencing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(7):956–960, 2004. 10

[Zil+10] F. Zilly, M. Müller, P. Eisert, and P. Kauff. Joint Estimation of Epipolar Geometry and Rectification Parameters using Point Correspondences for Stereoscopic TV Sequences. In *Proc. Int. Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT)*, 2010. 79

[ZYX11] J. Zhu, R. Yang, and X. Xiang. Eye Contact in Video Conference via Fusion of Time-Of-Flight Depth Sensor and Stereo. *3D Research*, 2(3):1–10, 2011. 11

Selbstständigkeitserklärung

Hiermit erkläre ich, die vorliegende Dissertation mit dem Titel *Real-time 3D-based Virtual Eye Contact for Video Communication* selbstständig und nur unter Verwendung der angegebenen Hilfen und Hilfsmittel angefertigt zu haben.

Ich habe die Arbeit nicht bereits an einer anderen Universität eingereicht und besitze keinen Doktorgrad im Fach Informatik.

Die Promotionsordnung der Mathematisch-Naturwissenschaftlichen Fakultät vom 30.06.2014, veröffentlicht im Amtlichen Mitteilungsblatt der Humboldt-Universität zu Berlin Nr. 126/2014 am 18.11.2014, ist mir bekannt.

Berlin, den 27. Februar 2018

Wolfgang Waizenegger